Universidade de Aveiro Departamento de Física
2008

ANA RITA SECCA DA FONSECA

DESENVOLVIMENTO DE UM EYE TRACKER DE BAIXO CUSTO

Universidade de Aveiro Departamento de Física
2008

ANA RITA SECCA DA FONSECA

DESENVOLVIMENTO DE UM EYE TRACKER DE BAIXO CUSTO

dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Física, realizada sob a orientação científica do Dr. Fernão Vístulo de Abreu, Professor Auxiliar do Departamento de Física da Universidade de Aveiro

o júri

presidente

Prof. Dr. João Lemos Pinto
professor catedrático do Departamento de Física da Universidade de Aveiro
Prof. Dr. Fernão Vístulo de Abreu
professor auxiliar do Departamento de Física da Universidade de Aveiro
Prof. Dr. João Manuel R. S. Tavares
professor auxiliar com nomeação definitiva da Faculdade de Engenharia da Universidade do Porto, Departamento de Engenharia Mecânica e Gestão Industrial

palavras chave

segmentação de imagem, técnicas de optimização funcional, processamento visual, eye tracker, reflexão corneal, pupila, redes neuronais

resumo

Neste trabalho apresenta-se o desenvolvimento de um eye tracker de baixo custo. São discutidos os aspectos experimentais considerados no seu desenvolvimento, tais como: a escolha da câmara, da lente e do aparato experimental. São analisados os métodos de segmentação de imagem utilizados. Com vista à calibração do eye tracker usou-se uma rede neuronal e a sua implementação é explicada. Por fim, é avaliada a performance do eye tracker.

keywords Image segmentation, functional optimization methods, visual processing, eye tracker, corneal reflection, pupil, neural networks

abstract The following work describes the development of a low cost eye tracker. Several experimental aspects are discussed such as the camera, the lens and the experimental set. There is an analysis of the image segmentation methods used. For calibration of the eye tracker, a neural network is considered and its implementation is explained. Finally, the eye tracker performance is assessed.

DEVELOPMENT OF A LOW COST EYE TRACKER

1. INTRODUCTION

An eye tracker can be used for several applications. Mainly, it is based on the premise that the pupil movement and position give information about the visual processing mechanisms. In spite of the eye system (e.g. pupil, cornea, retina, etc.) being well known it is not straightforward the way how the brain manages and uses the information it receives. Describing this type of processing can help us to better understand how the brain works. This kind of information is valuable for physicists, neurologists, marketing professionals, biologists, psychologists, etc.

Developing an eye tracker uses concepts from physics within an engineering approach. Hence, some mathematical models needed to be understood to perform image segmentation and robust methods had to be implemented. The eye tracker development involved building a head fixation apparatus, choosing lens, choosing a camera and choosing an infra red light to illuminate the eye. It was also necessary to develop an experimental procedure such as finding a proper positioning of the subject in relation to the camera or determine proper conditions of operation of the eye tracker (e.g. ambience illumination, the eye illumination). Additionally, software had to be developed in order to implement segmentation and calibration procedures.

In this work, I start by presenting some concepts and methods for image segmentation. These methods are essential for the eye tracker operation. Consequently, I engage in a detailed analysis of the mathematical frameworks commonly used to solve segmentation problems. This analysis points out that each method has its own advantages and problems, being more successful in some situations than others. Our work was initially driven by the work of De Santis and Iacoviello where an *Optimal segmentation of pupillometric images for estimating pupil shape parameters* was presented. They used the active contours segmentation method. However, the application of this method to typical eye images obtained with the Pixelink camera showed some limitations. Therefore, an alternative method was developed. We will discuss its implementation, as well as its limits and advantages.

After determining the segmentation method, a calibration procedure was developed. The calibration process establishes a relation between the eye images acquired and the place the subject is looking at. There is a relation between the movement of the pupil, the movement of a light spot reflected on the cornea from the infra red illuminant, and the point of gaze. I used a neural network to estimate it. Finally, I present some details about the camera, lens, infra red lamp used and also some characteristics of the head fixed apparatus.

2. IMAGE SEGMENTATION

2.1 Image versus information:

An image can be a source of information. For instance, medical images are sources of anatomical and functional information; visual surveillance images are sources of information about a specific event; industrial processes images are sources of information about quality control and measurements. Nevertheless, in most of these images the information is not clear and ready to be interpreted by humans or by machines. In order to extract information from these images, several methods can be used. Without them it would be very difficult to understand and manipulate information. These processes use images as input in order to output information about specific objects to which these images relate.

There are three main areas in which these processes can be categorized: image processing, image analysis and image understanding.

- **Image processing** is related to every process that can be applied to the image in order to that information from its objects becomes clear and evident. The input of these processes is the image and their output is also an image. Examples could be image filtering to decrease noise levels, or increasing the difference between the highest and the lowest levels of pixel's intensity to increase contrast.
- **Image analysis** includes all the processes which involve the extraction of data from images by automatic or semi-automatic methods. For example, the process of image segmentation which consists of isolating specific objects in the image. The efficacy of this process has an important impact in image understanding. When objects are isolated with success it is easier to extract accurate information.
- **Image understanding** involves all the processes that assess what kinds of objects are present in the image. These processes reveal if a specific object is a person, a circle or a blood vessel. After image segmentation, objects are isolated allowing the determination of attributes such as area, perimeter or form. Using a data base of known objects, attributes can be compared so that we are able to classify the objects in the image. Basically, these processes outputs are high-level descriptions of the objects in the image. This can improve the efficiency of automated responses.

Here we will focus particularly on processing and analyzing 2D images.

2.2 Main Concepts

An analog image is a 2-dimensional light intensity function, $f(x,y)$, where x and y are spatial coordinates and the value of f at x,y is proportional to the brightness of the image at that point. A digital image is an image $f(x,y)$ that

has been discretized both in spatial coordinates and in brightness. This discrete domain is a matrix where the location of each element corresponds to the position of each pixel and where each element is the value of intensity of that pixel (see Fig.1). Computationally, that matrix will have dimension given equal to the image resolution. For RGB images, there are three matrices each associated with one colour (Red, Green and Blue). For greyscale images, there is a single matrix. In both cases, the value for each matrix element ranges between 0 and 255. In binary images two intensities are allowed: 0 for black and 1 for white. The information contained in the matrix is extracted using mathematical operations. A matrix can be manipulated using point, local or global operations (see Fig.2):

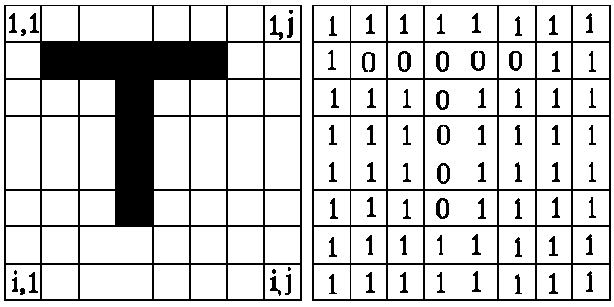


Figure 1 On the left we can see the letter T on a white background. On the right we can see the corresponding matrix. This matrix as a resolution 8x8.

- In **point operations** a specific pixel output value depends only on the input value of that same pixel.
- In **local operations** a specific pixel output value depends on the input values of the neighbouring pixels.
- In **global operations** a specific pixel output value depends on all values of the input image.

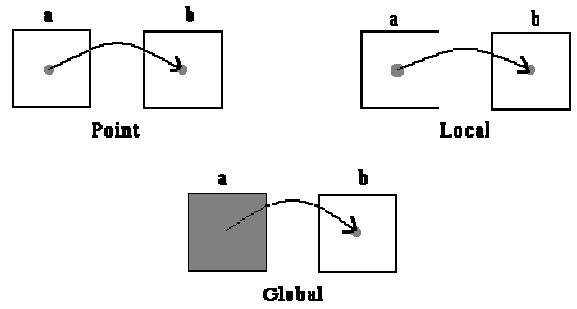


Figure 2 Different types of operations with which to manipulate matrices.

These mathematical operations can include the determination of mean intensity values, the determination of gradients, the smoothing of groups of pixels, making use of the intensity histogram to increase contrast or to apply filters, binary arithmetic operations to invert intensities or to eliminate objects. With these operations we can implement algorithms in order to approximate a machine automated process to a brain natural process (image understanding).

2.3 Segmentation Process

An image is constituted by a background and a foreground. The foreground is the group of objects that we want to isolate. Often, there are regions of pixels with the same intensity. Regions with distinct properties can define different objects. Each object has boundaries that can be accurately defined or not, which can be occluded by other objects or not. Therefore, the

segmentation process intends to copy some mechanism that our brain uses while we are staring at an image. We gather pixels which have the same intensity. We associate abrupt gradients to boundaries and homogeneity to regions. More rigorously, the goal of a segmentation process is to partition the image domain W into sub-domains W_i that constitute the objects. These have uniform parameters of intensity, colour, texture and crisp, regular boundaries. There are many frameworks that can solve the segmentation problem. For our work we used a method generally designated as *active contours without edges*.

2.4 Active Contours

In 1988, Kass, Witkin, and Terzopoulos[11] introduced a method of segmentation that uses the motion of a curve. The idea is to initialize a curve in an image and let this curve move until it conforms to the contour of a particular object. The curve's motion depends on a driving force that directs the curve to the contours of the objects. This driving force can be associated to a potential energy defined by a functional. This potential energy should reach its minimum near the object's contours. Thus, an energy is associated to the segmentation of an image. Following this framework, image segmentation turns into an optimization problem that can be solved using a Variational Method. The equation that defines the potential energy determines the motion of the curve and must converge to a local or a global minimum. In image segmentation, this potential energy is defined by a functional that depends on global and/or local properties of the image. These properties impose constraints to the motion of the curve. In active contours, the boundaries of a specific object can be determined by edge detection (local properties) or region detection (global properties). In edge-based segmentation, a transition between two sub-regions can be determined on the basis of discontinuities alone, if those sub-regions are sufficiently uniform. For region-based segmentation it is assumed that each sub-region is sufficiently uniform and distinct from other regions so that it can be considered an object. Hence, if we have an image where there are homogeneous regions with high gradients on the boundaries of objects, edge detection methods should be used and if we have an image with low gradients or a high degree of noise, region detection methods may achieve better results.

A curve can be defined by an explicit or implicit geometric form. In image segmentation, there is a variational method that uses an explicit definition of the contours of the objects – the *snakes method* – and other that uses an implicit definition of that curve – the *level set method*. Here we only present a brief introduction to the *snakes* method since our work was based on *level sets*.

2.5 Snakes Method

Consider a contour defined parametrically by the arc length s as:

$$C(s) \equiv \{(x(s), y(s)) : 0 \le s \le L\} : \Re \to \Omega \text{ (1)}$$

where L denotes the length of the contour C and Ω denotes the entire domain of the image $I(x, y)$. An energy function $E(C)$ can be defined on the contour, having to contributions:

$$E(C) = E_{\text{int}} + E_{ext} \text{ (2)}$$

where E_{int} and E_{ext} denote respectively the *internal energy* and *external potential energy* functions. The internal energy function uses the derivatives of the contour C to account to its smoothness. The regularity of the contour is related with the differentiability of C. A common choice for the internal energy is given by:

$$E_{\text{int}} = \int_0^L \alpha \left| C'(s) \right|^2 + \beta \left| C''(s) \right|^2 ds \text{ (3)}$$

Here α controls the tension of the contour, and β controls the rigidity of the contour. The external potential energy term determines the criteria for the contour evolution, depending on the image $I(x, y)$, and can be defined as:

$$E_{\text{int}} = \int_0^L E_{img}(C(s)) ds \text{ (4)}$$

where $E_{img}(x, y)$ denotes a scalar potential function defined on the image plane, so that the local minimum of E_{img} attracts the contour to the edge. The function E_{img} depends on local gradients of intensities, usually associated with edges [11]. For example, E_{img} may choose to be $E_{img}(x, t) = c \left| \nabla G_\sigma I(x, y) \right|$, where c is a suitably chosen constant, and G_σ denotes a Gaussian smoothing filter of standard deviation σ. Internal and external energies where defined in a continuous domain but, as mentioned, images are represented by matrices. Hence, these energies must be defined on a discrete domain. This means that contour C will not be a continuous curve but a connected set of points. The snake points are first initialized, then for the new iteration a new value is calculated, to minimize the energy. By connecting these snake points the evolved contour is the displayed. This procedure is repeated until all snake points stop in the optimal coordinates associated with the local minimum of energy. The main problem associated with the *snakes* method is its incapability of changing the topology as it does not allow splitting or merging into new boundaries. The

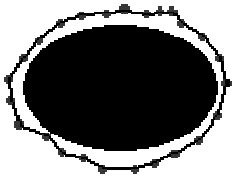


Figure 3 An example of a contour in the snakes method.

implicit formulation of curves by *level sets* was presented as an alternative to the *snakes* method because it could change the topology of the initial curve.

2. 6 Level sets

The *level sets* method was popularized by S. Osher and J.A. Sethian [16] and nowadays it is still very popular as an image segmentation framework. They introduced in the energy functional (Eq. 21) the implicit formulation of a curve.

A level set of a function ϕ defines implicitly the curve that is to be conformed to the contours of objects. This level set, $\phi(x, y, t) = 0$, evolves as a two dimensional curve embedded in a hypersurface (see Fig. 4). The function ϕ is defined as follows:

$$\begin{cases} \phi(x, y, t) > 0, \forall x, y \in \Omega_1 \\ \phi(x, y, t) = 0, \forall x, y \in C \\ \phi(x, y, t) < 0, \forall x, y \in \Omega_2 = \Omega / \Omega_1 \end{cases} \quad \textbf{(5)}$$

The initial curve corresponds to the set of points for which $\phi(x, y, t = 0) = 0$. This initial zero level set. At each time a contour C can be defined as satisfying:

$$C \equiv \{(x, y) : \phi(x, y) = 0\}, \forall (x, y) \quad \textbf{(6)}$$

where Ω is the image domain and x, y is a position in the image. The advantage of using the zero level is that a contour can be defined as the border between a positive area and a negative area. Hence, the contours can be identified by just checking the sign of $\phi(x, y)$. Parametrically, $\phi(x, y, t) = 0$ can be defined as:

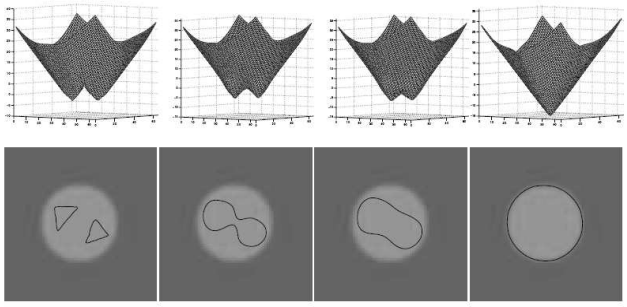


Figure 4 The fist row of images show the evolution of the surface ϕ. The second row of images show the evolution of the zero level set towards the boundaries of the object. As can be observed, the two initial contours merged to form just one contour, that is, there was a change in topology.

$$\phi(C(t), t) = 0 \quad \textbf{(7)}$$

Differentiating the previous equation using the chain rule, it follows:

$$\frac{\partial \phi}{\partial t} + |\nabla \phi| \cdot \frac{\partial C}{\partial t} = 0 \quad \textbf{(8)}$$

The evolution equation of the level set function

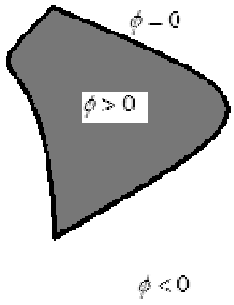


Figure 5 The function ϕ defines two domains and one boundary.

ϕ can be written in the general form known as the level set equation:

$$\frac{\partial \phi}{\partial t} + V|\nabla \phi| = 0 \quad \textbf{(9)}$$

where the function V is the speed function along the normal direction. In Osher and Sethian pioneer work this was defined as:

$$V = \left(\nu + \varepsilon \cdot div\left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right) \quad \textbf{(10)}$$

In traditional level set methods, this kind of curve motion is called a mean curvature flow because of the term $div\left(\frac{\nabla \phi}{|\nabla \phi|} \right)$. This term is the mean curvature of the level set function and it controls the regularity of the contour. ν denotes a constant speed term that pushes or pulls the contour. ε establishes the balance between regularity and robustness. Therefore, comparing Eq.8 and Eq.9 we can observe that:

$$\frac{\partial C}{\partial t} = V \quad \textbf{(11)}$$

This equation describes the motion of an interface in a direction normal to itself with a known speed function V.

After Osher and Sethian, Chan and Vese [3] defined the deformation of the contour describing another relation between the $\phi(x, y)$ and the contour C. A function $L_\varepsilon(\phi(x, y))$ approximates the length of the contour:

$$|C| \approx L_\varepsilon(\phi(x, y)) = \int_\Omega |\nabla H_\varepsilon(\phi(x, y))| dxdy \quad \textbf{(12)}$$

where H_ε is the regularized Heaviside function:

$$H_\varepsilon(\phi(x, y)) = \lim_{\varepsilon \to 0} \frac{1}{2}(1 + \frac{2}{\pi} \tan^{-1}(\frac{\phi(x, y)}{\varepsilon})) \quad \textbf{(13)}$$

and analogously, the regularized Dirac function can be defined by:

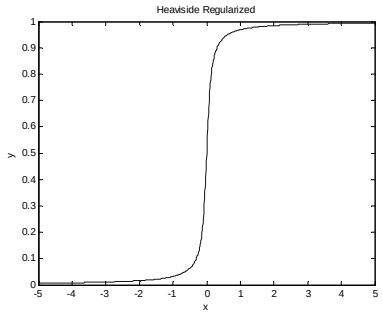


Figure 6A The regularized Heaviside function with $\varepsilon = 0.1$.

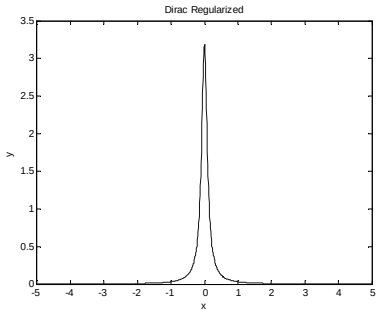


Figure 6B The regularized Dirac function with $\varepsilon = 0.1$.

$$\delta_\varepsilon(\phi(x,y)) = \lim_{\varepsilon\to 0}\frac{1}{\pi}\left(\frac{\varepsilon}{\varepsilon^2+\phi^2(x,y)}\right) \textbf{(14)}$$

Following Eq.12, Eq.13 and Eq.14 we have:

$$\nabla H_\varepsilon(\phi(x,y)) = \left(\frac{\partial H_\varepsilon}{\partial x}, \frac{\partial H_\varepsilon}{\partial y}\right) = \left(\frac{\partial H_\varepsilon}{\partial \phi}\frac{\partial \phi}{\partial x}, \frac{\partial H_\varepsilon}{\partial \phi}\frac{\partial \phi}{\partial x}\right) \textbf{(15)}$$

So that,

$$\frac{\partial H_\varepsilon}{\partial \phi}\frac{\partial \phi}{\partial x} = \frac{1}{\pi}\frac{1}{\varepsilon}\frac{1}{1+\frac{\phi^2(x,y)}{\varepsilon^2}}\frac{\partial \phi}{\partial x}$$

$$\frac{\partial H_\varepsilon}{\partial \phi}\frac{\partial \phi}{\partial y} = \frac{1}{\pi}\frac{1}{\varepsilon}\frac{1}{1+\frac{\phi^2(x,y)}{\varepsilon^2}}\frac{\partial \phi}{\partial y} \qquad \textbf{(16)}$$

$$\nabla H_\varepsilon(\phi(x,y)) = \frac{1}{\pi}\frac{\varepsilon}{\varepsilon^2+\phi^2(x,y)}\left(\frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y}\right) = \delta_\varepsilon(\phi(x,y))\nabla\phi(x,y)$$

Substituting $\nabla H_\varepsilon(\phi(x,y))$ in Eq.12, we get

$$|C| \approx \int_\Omega \delta_\varepsilon(\phi(x,y))|\nabla\phi(x,y)|dxdy \textbf{(17)}$$

Another function of $\phi(x,y)$ can be used to approximate the area of the contour:

$$|w| \approx \int_\Omega H_\varepsilon(\phi(x,y))dxdy \textbf{(18)}$$

As we can observe from the Eq.17, the length of C is approximated by the sum of all the norms of the gradient of $\phi(x,y)=0$ for each point (x, y). The area of the contour is approximated by the sum of all points that are inside the contour.

The minimization of this length leads to the following evolution equation:

$$\frac{\partial \phi}{\partial t} = \delta(\phi(x,y))\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right) \textbf{(19)}$$

In comparison with the *snakes* method, the main advantage of *level sets* method is that it handles naturally cavities, concavities, convolution, splitting or merging on contours. The initial curve can be a closed contour and evolve to two closed contours or vice versa. Due to this fact, it is possible an arbitrary initialization of the contours which excludes the need for having previous information about the objects. It also allows the segmentation of objects with discontinuous edges, that is, it can handle contours which are not regular.

2.7 Edge Based

For an Edge-Based approach an additional term is introduced in the snake model functional. This term depends on the gradient of the intensities. Within this approach, the boundary of an object takes place when the gradient is maximal. It estimates whether a given point in the image separates two distinct and uniform areas, assuming that this point pertains to the object's boundary. Therefore, the stopping criterion for the minimum of energy is a gradient maximum. As follows:

$$E = \int_0^L (\alpha \left| C'(s) \right|^2 + \beta \left| C''(s) \right|^2) ds - \lambda \int_0^L \left| \nabla u_0 (C(s)) \right|^2 ds \quad \textbf{(20)}$$

An Edge Based active contour model uses local properties in order to constrain the movement of the curve. As it evaluates local minimums it is very susceptible to noise.

2.8 Region Based

A Region Based active contour model uses global properties in order to constrain the movement of the curve. These models evaluate the uniformity of regions in the image, considering that a uniform region is associated with an object and it also evaluates the smoothness as a parameter of regularity. For a Region Based model, the functional has terms that evaluate the uniformity of parameters like intensity, colour, texture, etc. The segmentation energy for region based active contour models is described by the functional of Mumford-Shah.

2.9 Mumford-Shah Functional

The application of the Mumford-Shah functional proposes the following solution to the segmentation problem: given an observed image u_0, find a decomposition ω_i of ω, such that the new segmented image u then varies smoothly within each ω_i, and discontinuously across the boundaries of ω_i. The Mumford-Shah functional can be written as:

$$E(u, C) = \sum_i \lambda \int_{\Omega_i} \left| u_o(x, y) - c_i \right|^2 dxdy + \nu \left| C \right| \quad \textbf{(21)}$$

The minimization process depends on c_i, which is the mean intensity in the region ω_i, which is term associated with homogeneity; and depends on C characteristics, where C is a set of curves in ω, i. e., it is the set of boundaries of the objects.

In 2001, Chan and Veese[5] developed a new approach to *active contours* using the Mumford-Shah functional instead of the classical approach based on

the gradient. They also used the *level set* method formulation of the model. Their functional integrates the terms that evaluate the smoothness within a region and also two more terms. These terms establish constraints on the length of object's boundary and its area. The length constraint can help eliminating isolated points near the boundaries of the objects whereas the area constraint may eliminate islands of points associated with noise. The full functional becomes.

$$E(\phi(x,y,t),c_1,c_2)=\lambda \int_{inside(\phi)} |u_o(x,y)-c_1|^2 dxdy+\lambda \int_{outside(\phi)} |u_o(x,y)-c_2|^2 dxdy+ \\ +\mu\cdot Area(\phi<0)+\nu\cdot Length(\phi=0) \quad \textbf{(22)}$$

Now that all these concepts were introduced we can discuss how this segmentation method can be implemented and what kind of results can be obtained.

2.10 Discrete analysis

First we need to define the functional E (Eq. 22) in the discrete space:

$$E(\phi(i,j,t),c_1,c_2)=\lambda\sum_{i,j} H(\phi(i,j,t))|u_0(i,j)-c_1|^2 + \\ +\lambda\sum_{i,j}(1-H(\phi(i,j,t)))|u_0(i,j)-c_2|^2 + \\ +\mu\sum_{i,j} H(\phi(i,j,t))+\nu\cdot Length(\phi=0) \quad \textbf{(23)}$$

where $H(\phi)$ is the Heaviside function:

$$H(\phi)=\begin{cases}1, \phi \geq 0\\ 0, \phi < 0\end{cases} \quad \textbf{(24)}$$

and $\delta(\phi)$ is the derivative of $H(\phi)$, i. e., the Dirac function:

$$\delta(\phi)=\begin{cases}1, \phi=0\\ 0, \phi\neq 0\end{cases} \quad \textbf{(25)}$$

As we are working in a discrete level $\phi(i,j,t)$ is only rarely equal to zero. Therefore, for a discrete level we must work with the regularized versions of $H(\phi)$ and $\delta(\phi)$ (Eq. 13 and Eq.14). To determine the length of the contour we define an auxiliary function, $\gamma(\phi(i,j))$, that evaluates if a pixel is a boundary point. $\gamma(\phi(i,j))$ was defined by De Santis and Iacoviello[7] as:

$$\gamma(\phi(i,j))=H(3-\rho(\phi(i,j)))[1-\delta(\rho(\phi(i,j)))] \quad \textbf{(26)}$$

and $\rho(z(i,j))$ is defined as:

$$\rho(z(i,j)) = \sum_{-1}^{1} \left[(H(z(i+l,j)) - H(z(i,j)))^2 + (H(z(i,j+l)) - H(z(i,j)))^2 \right] \quad (27)$$

The function $\rho(\phi(i,j))$ evaluates how many times the signal of $\phi(i, j, t)$ changes within the close neighbours of a particular pixel (see Fig.7). In such way that:

- if $\rho(\phi(i,j)) = 3$, then one of the neighbours has the same signal for $\phi(i, j, t)$ as the pixel (i, j). Hence, (i, j) belongs to the boundary.
- if $\rho(\phi(i,j)) = 2$, then two of the neighbours have the same signal for $\phi(i, j, t)$ as the pixel (i, j). Hence, (i, j) belongs to the boundary.
- if $\rho(\phi(i,j)) = 1$, then three of the neighbours have the same signal for $\phi(i, j, t)$ as the pixel (i, j). Hence, (i, j) is a boundary pixel.
- if $\rho(\phi(i,j)) = 4$, then all four neighbours have the same signal for $\phi(i, j, t)$ among them, but different signal from the pixel (i, j). Hence, (i, j) is an isolated pixel.
- if $\rho(\phi(i,j)) = 0$, then all five pixels have the same signal for $\phi(i, j, t)$. Hence, (i, j) is an interior pixel.

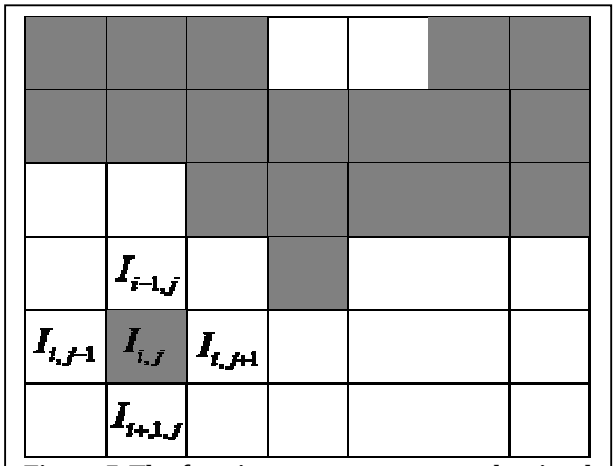


Figure 7 The function $\rho(\phi(i,j))$ assesses the signal change of the function $\phi(i, j)$ for the neighbours of the pixel (i, j).

Observing the function $\gamma(\phi(i,j))$, we can see that:

- if $\rho(\phi(i,j)) = 4$ then $H(3 - \rho(\phi(i,j))) = 0$ otherwise $H(3 - \rho(\phi(i,j))) = 1$;
- if $\rho(\phi(i,j)) = 0$ then $[1 - \delta(\rho(\phi(i,j)))] = 0$ otherwise $[1 - \delta(\rho(\phi(i,j)))] = 1$;

In conclusion, we will use $\gamma(\phi(i,j))$ to count the number of pixels that belong to the boundary so that we can determine the length of the curve C.

In this way we can define the energy as:

$$E(\phi(i,j,t), c_1, c_2) = \lambda \sum_{i,j} H(\phi(i,j,t)) |u_0(i,j) - c_1|^2 + \lambda \sum_{i,j} (1 - H(\phi(i,j,t))) |u_0(i,j) - c_2|^2 + \\ + \mu \sum_{i,j} H(\phi(i,j,t)) + \nu \sum_{i,j} \gamma(\phi(i,j,t)) \quad (28)$$

The algorithm of the method can be divided in the following steps:

- the method initializes with an arbitrary field $\phi(x,y)$ defining an arbitrary curve, $\phi(x,y)=0$;
- the arbitrary curve defines two domains (inside and outside) depending on the sign of $\phi(x,y)$;
- the mean intensity inside each domain, c_1 and c_2, is recursively calculated so that the functional in Eq. 28 is minimized;
- given these values of c_1 and c_2, equation 28 is calculated and a recurrent equation is used to minimize $E(\phi(i,j,t),c_1,c_2)$ relatively to variations in $\phi(x,y)$. This changes $\phi(x,y,t)$ so that the curve defined by $\phi(x,y)=0$ tends to fit the segmented region.

De Santis and Iacoviello[7] also introduced a term in the functional that forces the convexity of the function E, so that $\phi(x,y,t)$ never diverges. The full functional then becomes:

$$\begin{aligned} E(\phi(i,j,t),c_1,c_2) &= \lambda\sum_{i,j} H(\phi(i,j,t))\left|u_0(i,j)-c_1\right|^2 + \\ &+ \lambda\sum_{i,j}(1-H(\phi(i,j,t)))\left|u_0(i,j)-c_2\right|^2 + \\ &+ \mu\sum_{i,j} H(\phi(i,j,t)) + \nu\sum_{i,j}\gamma(\phi(i,j,t)) + \frac{\alpha}{2}\sum_{i,j}\phi^2(i,j,t) \end{aligned} \quad \textbf{(29)}$$

The framework that we are using considers that the partition of the image in regions can be determined by an energy minimization which leads to:

$$\begin{cases} \dfrac{\partial E}{\partial c_1} = \dfrac{\partial(\lambda\sum_{i,j}H(\phi(i,j))\left|u_0(i,j)-c_1\right|^2}{\partial c_1} = -2\lambda(\sum_{i,j}H(\phi(i,j))u_0(i,j) - c_1\sum_{i,j}H(\phi(i,j))) = 0 \\ \dfrac{\partial E}{\partial c_2} = 0 \\ \dfrac{\partial E}{\partial \phi} = 0 \end{cases} \quad \textbf{(30)}$$

The minimization involves two steps. From the first two equations in Eq. 30, we obtain:

$$c_1 = \frac{\sum_{i,j}H(\phi(i,j))u_0(i,j)}{\sum_{i,j}H(\phi(i,j))} \quad \textbf{(31)}$$

and

$$c_2 = \frac{\sum_{i,j}(1-H(\phi(i,j)))u_0(i,j)}{\sum_{i,j}(1-H(\phi(i,j)))} \quad \textbf{(32)}$$

The constants c_1 and c_2 are mean of intensities of the domain 1 ($\phi > 0$) and domain 2 ($\phi < 0$), respectively. Then the partial derivative of the E in relation to ϕ leads to:

$$\begin{aligned}\frac{\partial E}{\partial \phi} &= \lambda \frac{\partial}{\partial \phi}\left(\sum_{i,j} H(\phi(i,j))\left|u_0 - c_1\right|^2\right) + \lambda \frac{\partial}{\partial \phi}\left(\sum_{i,j}(1-H(\phi(i,j)))\left|u_0 - c_2\right|^2\right) + \\ &+ \mu \frac{\partial}{\partial \phi}\left(\sum_{i,j} H(\phi(i,j))\right) + \\ &+ \frac{\partial}{\partial \phi}\left(\sum_{i,j}\gamma(\phi(i,j))\right) + \frac{\alpha}{2}\frac{\partial}{\partial \phi}\left(\sum_{i,j}\phi^2(i,j)\right) = 0\end{aligned} \quad \textbf{(33)}$$

which becomes:

$$\begin{aligned}\frac{\partial E}{\partial \phi} &= \lambda\left(\left|u_0 - c_1\right|^2\right)\delta(\phi(i,j)) - \lambda\left(\left|u_0 - c_2\right|^2\right)\delta(\phi(i,j)) + \\ &+ \mu\delta(\phi(i,j)) + \nu\frac{\partial}{\partial \phi}\left(\sum_{i,j}\gamma(\phi(i,j))\right) + \frac{\alpha}{2}2\phi(i,j) = 0\end{aligned} \quad \textbf{(34)}$$

where,

$$\frac{\partial}{\partial \phi}\left(\sum_{i,j}\gamma(\phi(i,j))\right) = \frac{\partial E}{\partial \gamma}\frac{\partial \gamma}{\partial \rho}\frac{\partial \rho}{\partial \phi} = 0 \quad \textbf{(35)}$$

so that,

$$\phi(i,j) + \frac{1}{\alpha}\left(\left[\lambda\left(\left(u_0 - c_1^2\right) - \left(u_0 - c_2^2\right)\right) + \mu\right]\delta(\phi(i,j)) + \nu\left(\frac{\partial E}{\partial \gamma}\frac{\partial \gamma}{\partial \rho}\frac{\partial \rho}{\partial \phi}\right)\right) = 0 \quad \textbf{(36)}$$

Finally,

$$\left(\lambda p(i,j)+\mu+\nu\sum_{l=-1}^{1}r(i,j,l)q(i,j)\right)\delta(\phi(i,j))+\alpha\phi(i,j)=0$$

$$r(i,j,l)=\left[\left(H(\phi(i+l,j))-H(\phi(i,j))\right)+\left(H(\phi(i,j+l))-H(\phi(i,j))\right)\right] \quad \textbf{(37)}$$

where $p(i,j)$ stands for:

$$p(i,j)=\left(u_0(i,j)-c_1\right)^2-\left(u_0(i,j)-c_2\right)^2 \quad \textbf{(38)}$$

$q(i,j)$ stands for:

$$q(i,j)=2\left[\delta(3-\rho(\phi(i,j)))\cdot(1-\delta(\rho(\phi(i,j))))-H(3-\rho(\phi(i,j)))s(i,j)\delta\left(\rho(\phi(i,j))\right)\right] \quad \textbf{(39)}$$

$$s(i,j)=\frac{2\rho(\phi(i,j))}{\varepsilon^2+\rho^2(\phi(i,j))}$$

Thus, the energy minimization has been transformed in the problem of finding the zeros of a function defined in a multidimensional space. A straightforward way of solving this problem is to use a fixed point method to find the recurrent equations. Having an equation of the type $G(\phi(i,j))=0$, we rewrite it as $G(\phi(i,j))+\phi(i,j)=\phi(i,j)$ which can be solved using the iterative method of searching for the fixed point (see Fig.8), which corresponds for the zero of the function.

2.11 Fixed point iteration method

If we consider ϕ_0 as an initial function, corresponding to initial arbitrary contour curves for which $G(\phi(i,j))=0$, we can write the recursive fixed point equation:

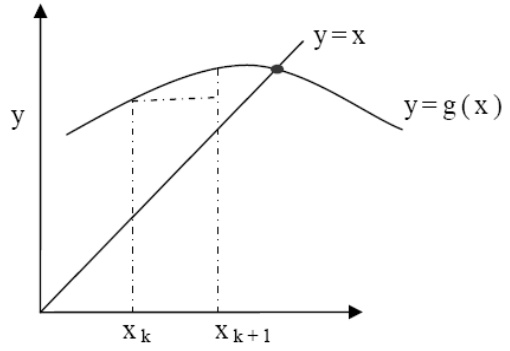


Figure 8 An example of the application of the fixed point iteration method

$$\phi^{n+1}=\phi^n\pm kG(\phi^n),\ n=0,1,..N \quad \textbf{(40)}$$

Here ϕ^n is the field of iteration n, k a constant whose sign and magnitude should be chosen so that the recursive equation converges to the fixed point, for which $\phi^{n+1}=\phi^n$, and hence $G(\phi^n)=0$. Determining $\phi^{n+1}=\phi^n$ solves the minimization problem. In this case, Eq.40 should be iterated until $\phi^{n+1}\left|\phi^{n+1}-\phi^n\right|\langle\varepsilon$, where ε is a tolerance limit fixed a priori. Hence, the recursive relation is defined as

$$\phi^{n+1}(i,j)=\phi^{n}(i,j)\pm k\left(\phi^{n}(i,j)+\frac{1}{\alpha}\left(\lambda p^{n}(i,j)+\mu+\nu q^{n}(i,j)\left(\sum_{l=-1}^{1}r(i,j,l)\right)\right)\delta\left(\phi^{n}(i,j)\right)\right) \quad (41)$$

The constant k should be chosen so that the method converges and quickly. Here was used $k=-0.5$. This method has a high cost on computational time because ϕ is updated for each pixel per iteration. For a matrix with 200x200 pixels this means that for 4 iterations 160000 updates are required. If per iteration are required 2ms of computational time, then a segmentation process could require 32 seconds, only for one image.

2.12 The application of the segmentation method

In this section, we discuss simple examples to illustrate the role of each parameter define in the functional. We varied the values of $\alpha, \lambda, \mu,$ υ separately and observed their impact in the segmentation. The segmentation method was applied to the an image (see Figure 9), and the convexity coefficient α was varied while $\lambda=100$, $\mu=0$, $\upsilon=0$ were kept fixed. For α:

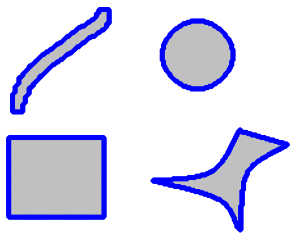


Figure 9 An image.

A. small ($\alpha=0.000001$), the segmentation was achieved after one iteration but ϕ diverged quickly reaching magnitude of the order of 10^{11} (see Fig.10);

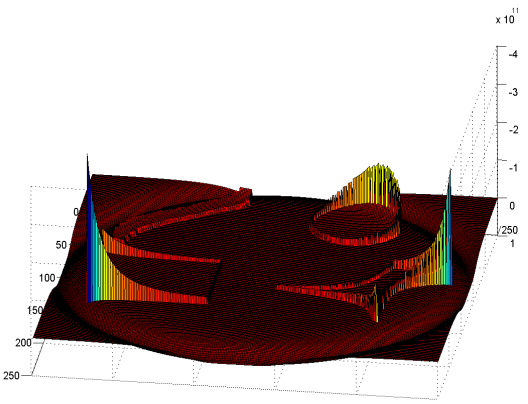


Figure 10 $\phi(i,j)$ **one four iteration when** $\alpha=100$.

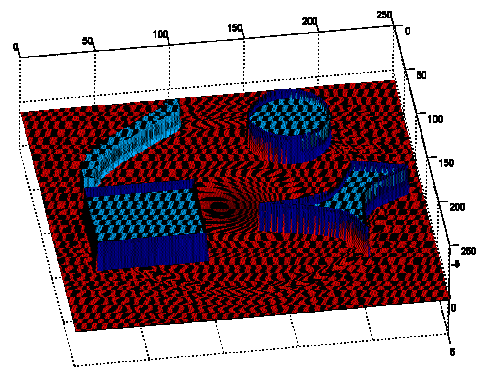


Figure 11 $\phi(i,j)$ **after four iterations when** $\alpha=100$.

B. large ($\alpha=100$) four iterations were required and the magnitude of ϕ remained finite, of the order of 10^{1} (see Fig.11).

If we compare the surface ϕ for both situations A and B:

- in A the difference of signal between the objects and the background is less accentuated. Nevertheless, this difference is enough to determine

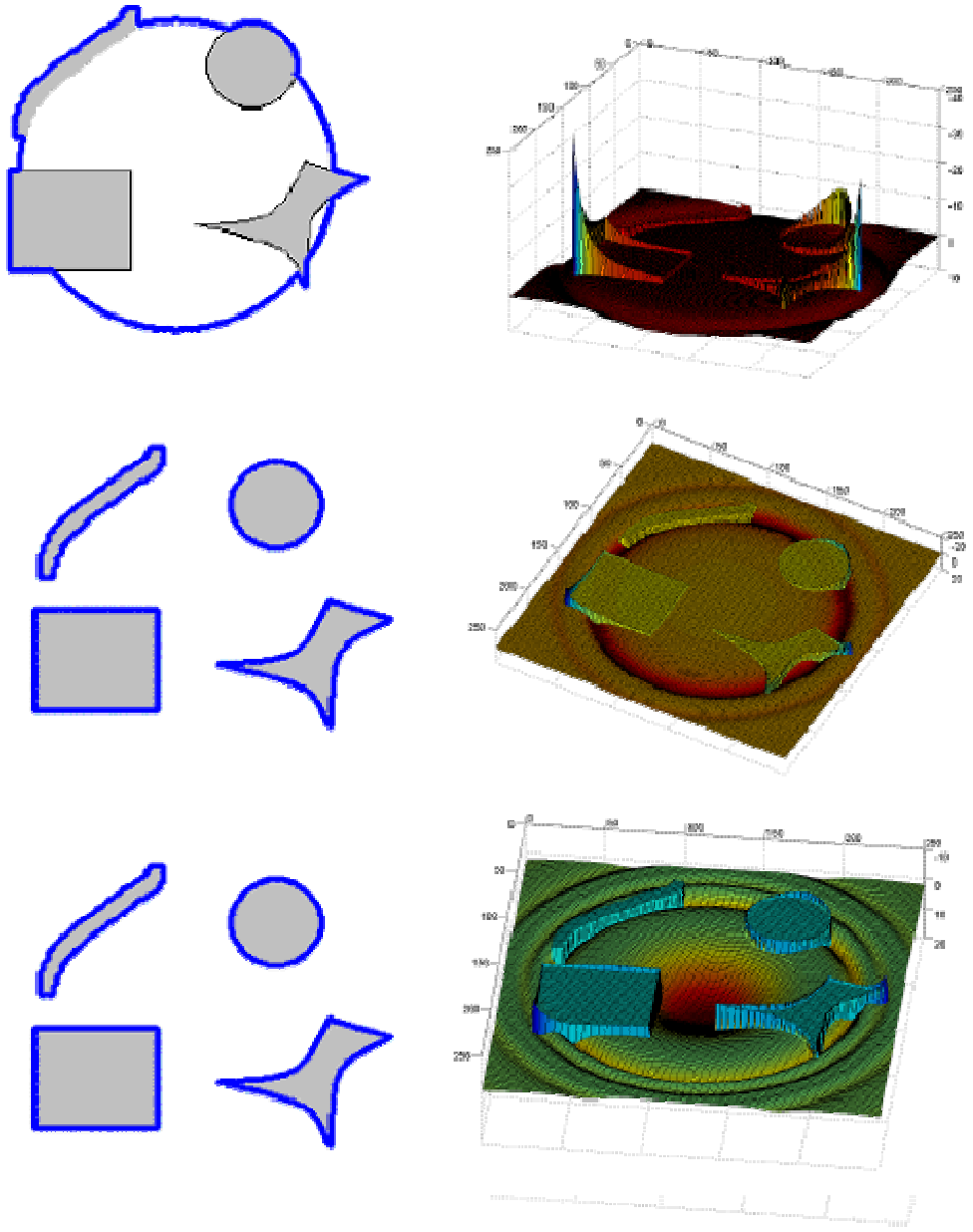


Figure 10 This sequence of images shows: on the left the segmentation of the image and on the right the corresponding surface ϕ for the application of the method with $\lambda = 1$. As we could observe, the segmentation of image is achieved in the second iteration. Despite of this, the part of the surface ϕ corresponding to objects only achieved an expressive difference from the background in the third iteration.

the boundaries of the objects;

- it can also be observed in situation A, that inside the objects there are different values for ϕ;

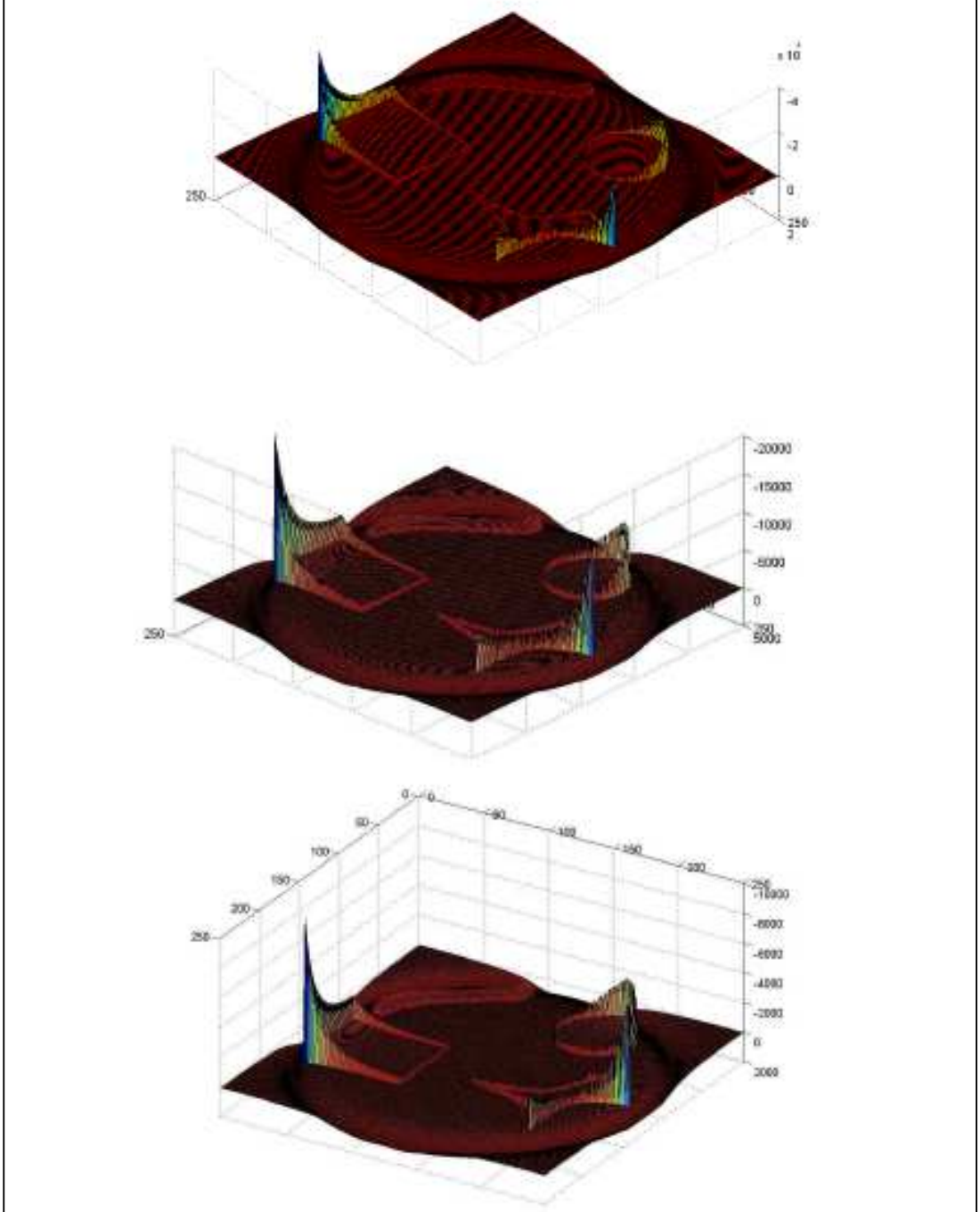


Figure 13 This sequence of images shows the surfaces of ϕ using $\lambda = 100$. The segmentation of the image was achieved in the first iteration (first image) but it was not achieved the convergence of ϕ as the surfaces of the 2nd and 11th iterations show.

- in B ϕ has a different signal for the objects and the background.
- there is also an accentuated difference in the magnitude of ϕ for both situations, of order of 10^{11} for situation A and 10^{0} for situation B.

Therefore, with this example we understand that the parameter α can be useful to avoid computational divergences.

Next, we used two different values for coefficient associated with homogeneity, $\lambda = 1$ (see Fig. 12) and $\lambda = 100$ (see Fig. 13), while $\alpha = 100$, $\mu = 0$, $\upsilon = 0$ (see). Further, when $\lambda = 0$ the segmentation is not achieved. This is predictable as this is the main term of the method associated with homogeneity. For cases like the one in Figure 9 this parameter and the one associated to convexity are enough to make this segmentation method to work properly.

In the next examples were used three values for the coefficient associated with the area $\mu = -100$, $\mu = -10$ and $\mu = 100$ while $\lambda = 0.01$, $\alpha = 1$, $\upsilon = 0$ where kept fixed (see Fig. 14). In case A there is a competition between both circles. Nevertheless, the black circle is the one that has the larger difference from the background's intensity. Although, λ has a low value it is the most important term in the segmentation so there is a balance between both parameters. In case B the segmentation for both circles was obtained. The value of μ is still negative but not too low so that the area must not be so small that it forces the exclusion of the grey circle. The last case, C, shows the maximization of the area for the parameter with positive signal. The segmentation was not successful.

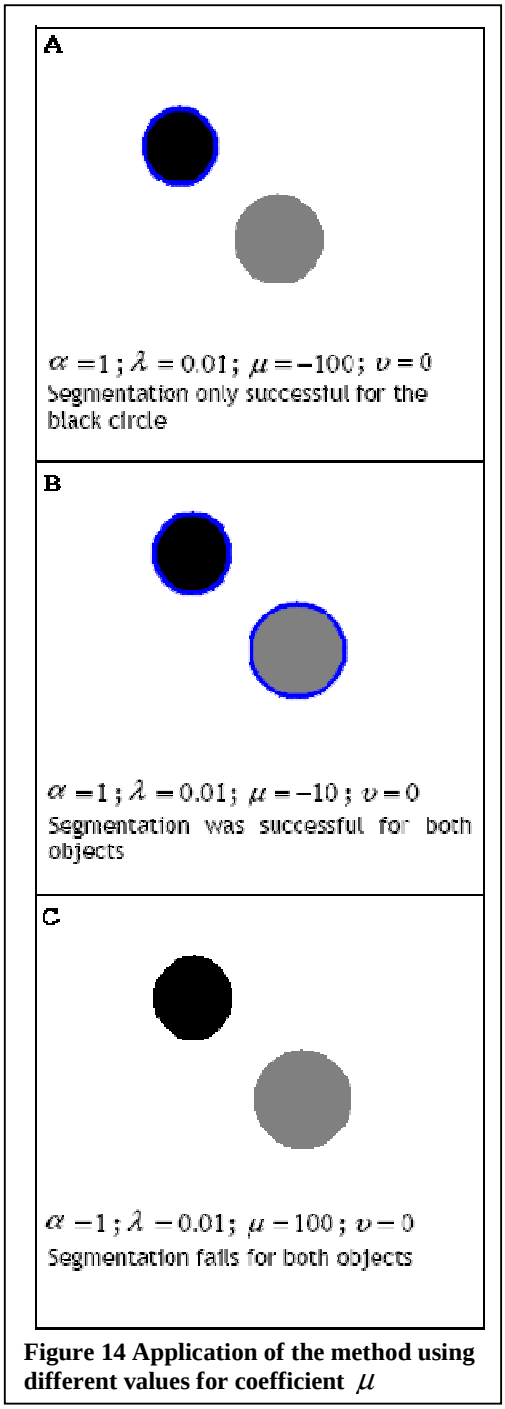


Figure 14 Application of the method using different values for coefficient μ

In the next examples we varied the value of the coefficient associated with the minimization of the contour length , $\nu = -1000$, $\nu = 10$ and $\nu = 100$ while $\lambda = 0.01$, $\alpha = 1$, $\mu = 0$ where kept fixed (see Fig.15). As can be observed the parameter minimizes the length if it is positive. Figure 15D shows that if the parameter ν is considerably negative it will introduce an error that does not disappear. In Figure15E, the segmentation was successful for both circles and the same happens in Figure15F, in spite of the ν increase. On the contrary, in G the grey circle is not segmented. The grey circle has a value of intensity closer

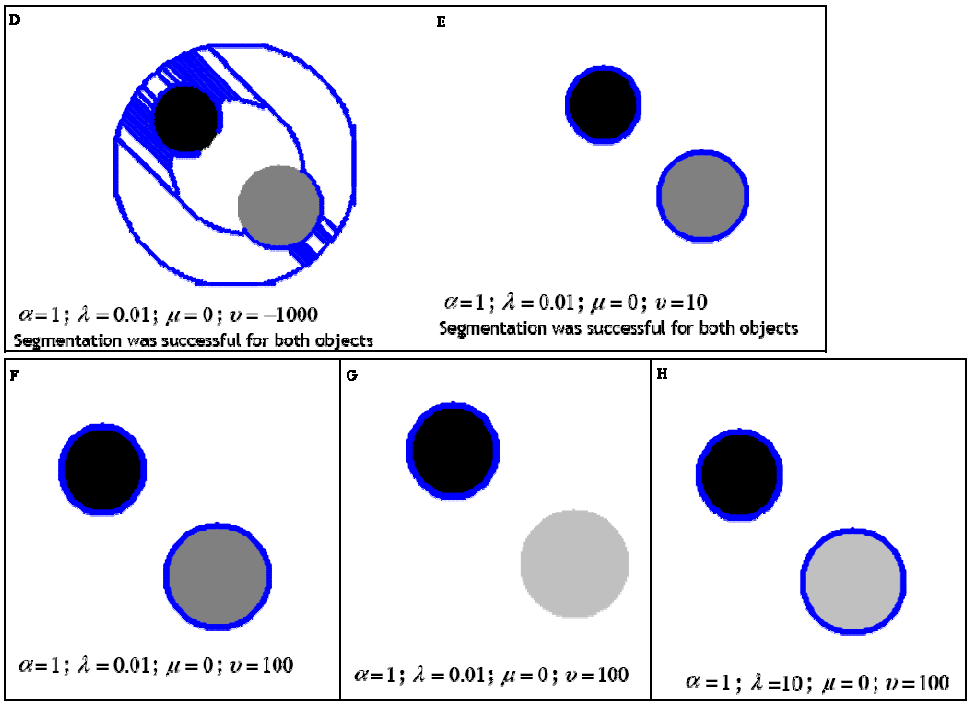


Figure 11 Application of the method using different values for coefficient ν .

to black values to white. So the strength of ν is not enough to exclude the grey circle and minimize the length. Hence in case G the light grey was not detected. This situation was reverted by giving more strength to the difference of intensity of the grey circle in relation to the background that is, increasing $\lambda = 10$ (situation H).

We conclude that a balance between parameters $\alpha, \lambda, \mu,\ \upsilon$ must exist. Summarizing:

- the parameter λ is the most important in the energy functional;
- the parameter α is necessary for the convexity of the of energy function;
- the parameter μ must be a negative, to minimize the area;
- the parameter υ must be positive, to minimize the length;

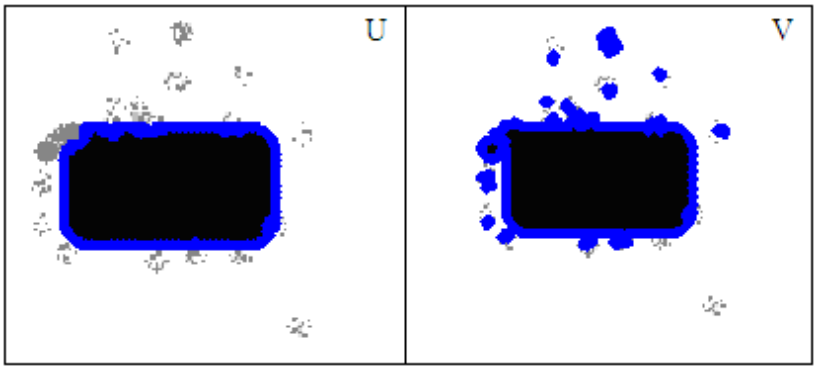


Figure 16 The application of the method with the parameters $\alpha = 0.01, \lambda = 1, \mu = -100, \nu = 100$ to images U and V. The image U and the image V are equal, excluding the colour of the points that simulate noise.

Nevertheless, all the parameters should be defined after a careful observation of the images to which the method is to be applied. For instance,

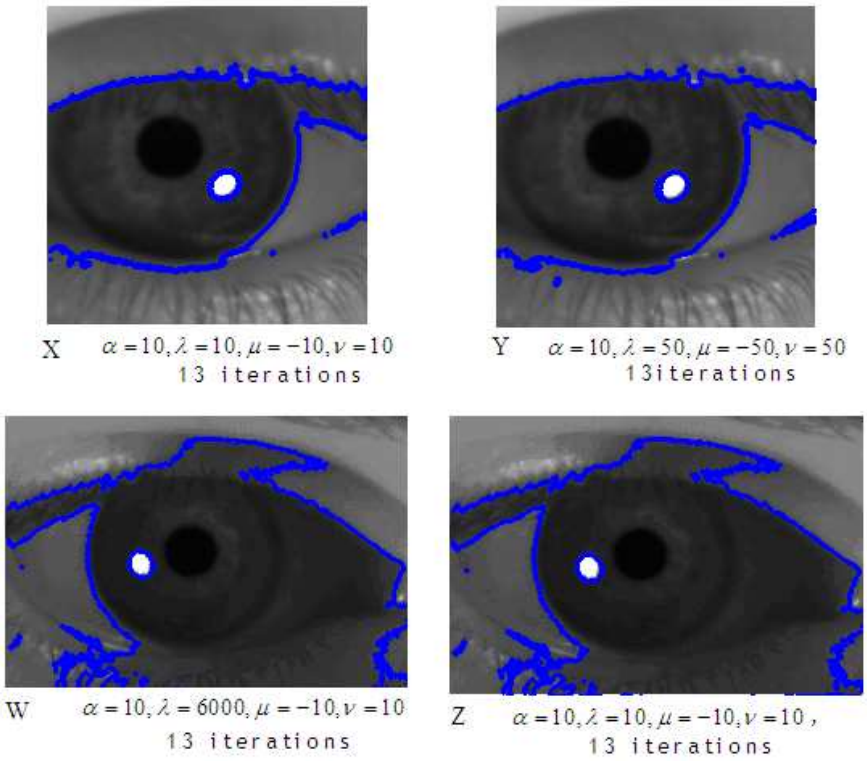


Figure 17 The application of the method to typical images, using different values for the parameters λ α,λ,μ,ν.

- if the image shows a high contrast between the background and the objects, the homogeneity term does not have to be very strong (small values for λ).
- if there is a low contrast between objects the homogeneity term has to be strong (λ big);
- if the boundaries of the objects are regular, the terms associated with regularity must no be very strong (μ,ν).
- if the image has noise the regularity terms should be strong (positive values for ν and negative values for μ); and what kind of noise (e.g. for isolated points, the term associated with the length minimization ν should be strong and for islands the term associated with the area minimization μ should be strong)

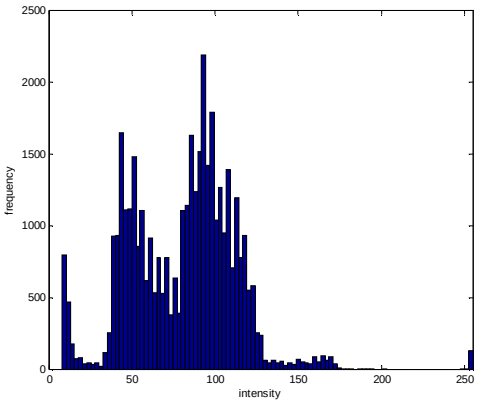


Figure 18 The histogram of intensities corresponding to the image in Figure 20X.

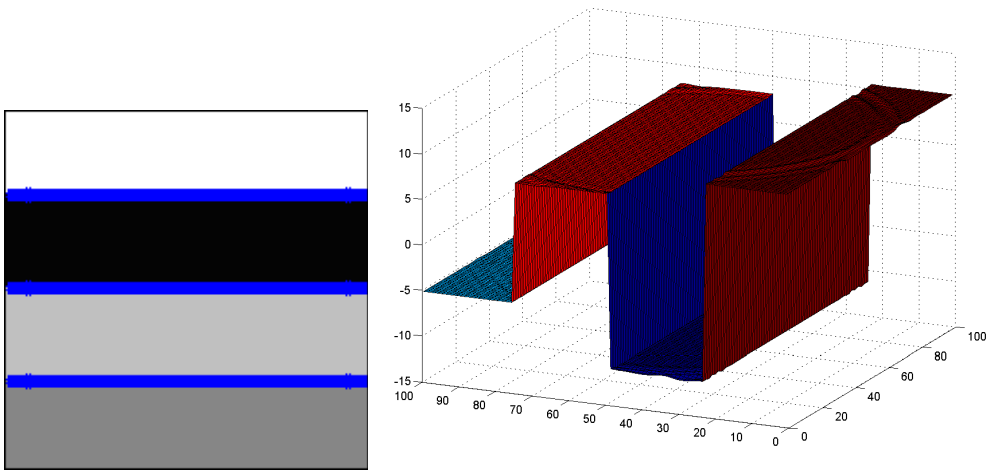


Figure 19A) On the left is the image used and the resulting segmentation. On the right is represented the surface ϕ. The parameters used were $\alpha = 10$, $\lambda = 10$, $\mu = 0$, $\nu = 0$

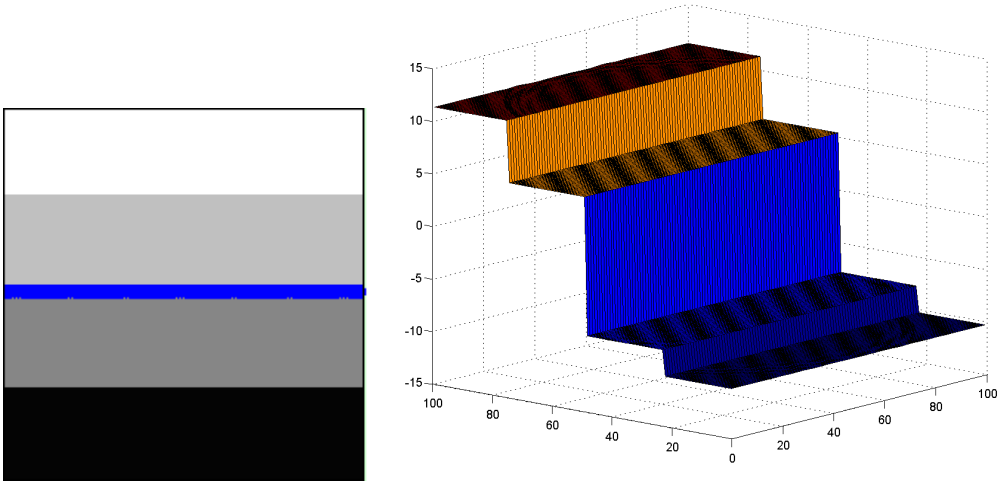


Figure 19B) On the left is the image used and the resulting segmentation. On the right is represented the surface ϕ The parameters used were $\alpha = 10$, $\lambda = 10$, $\mu = 0$, $\nu = 0$

Bearing these aspects in mind, we present next some examples for which the method either is successful or fails. In the first case in Figure 16 the segmentation was successful in spite of the 'noise' from light grey dots. However, in the second case where the 'noise' has the same colour as the object, and consequently the segmentation fails.

In Figure 17A, the segmentation was successful. In the image were detected four objects. We can see in the surface ϕ the fourth changes of signal between adjacent areas. In this case, the colours were interlaced by values of intensity. However, in the image in the Figure 17B, the segmentation fails. In this image only two objects were detected: the darker part of the image (black and dark grey) and the lighter part (white and light grey). Although the surface ϕ could be divided in four different areas, the change of signal is between the area that has a high mean of intensity and the area that has a low mean of intensity. Even with an increase of the parameter λ, which gives more strength to the

difference between the intensities of the different areas, we had was no success in performing a good segmentation.

The last example will be present in the typical eye images that were obtained. The object that we want to isolate from the rest of the image is the pupil. In order to increase contrast between the pupil and the iris, a infra red lamp is used. The pupil absorbs the infra red light so that it is, in principle, the darker object in the image. As we can observe for the different cases in Figure 18, the segmentation failed. In Figure 19 is represented the histogram of intensities of a typical image. For determining the centre of the CR it was applied a simple threshold on the image. As we can see there is a peak associated with the intensities above 250. In the case of the pupil the threshold does not work since there are several pixels with the same intensity has the pupil.

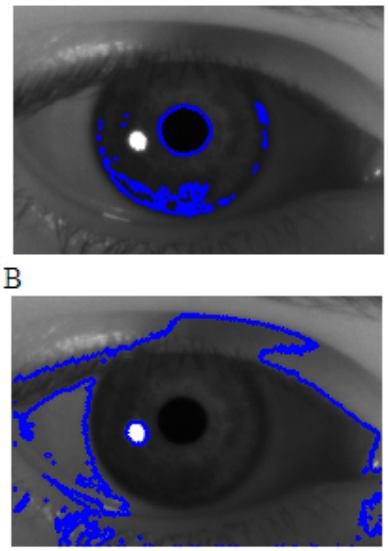


Figure 20 In A is represented the application of the method to typical images constraining the value of c_1 in the first iteration. In B is represented the method after thirteen iterations.

In order to improve the segmentation, we included the condition $c_1 = 0$ where c_1 is the mean of intensity of the domain 1, only for the first iteration of the segmentation method. In spite the first iteration showed better results, the surface ϕ evolved to the same kind of surface previously in the images of Figure 18. As the result obtained in Figure 20A was closer to a good solution, we tried to develop a complementary method which used that information to determine the center of the pupil. This method also finds an energy extremum based on energy minimization and it will be presented in section 2.14.

2.13 Image segmentation applied to optical illusions

The previous problems found in the application of the *active contours* method can be best illustrated when it is applied to optical illusions. The study of optical illusions from a scientific point of view has increased recently, since they allow a simple analysis of how our brain functions. In connection to segmentation problems it is really an interesting approach since it is known that

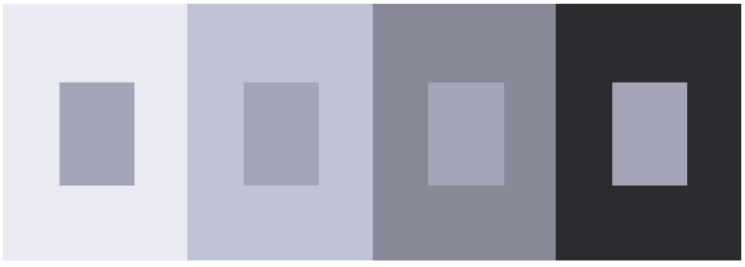


Figure 21 Although the central patch is exactly the same in all cases, simultaneous brightness contrast makes it appear to vary from dark to light as the background is changed from light to dark.

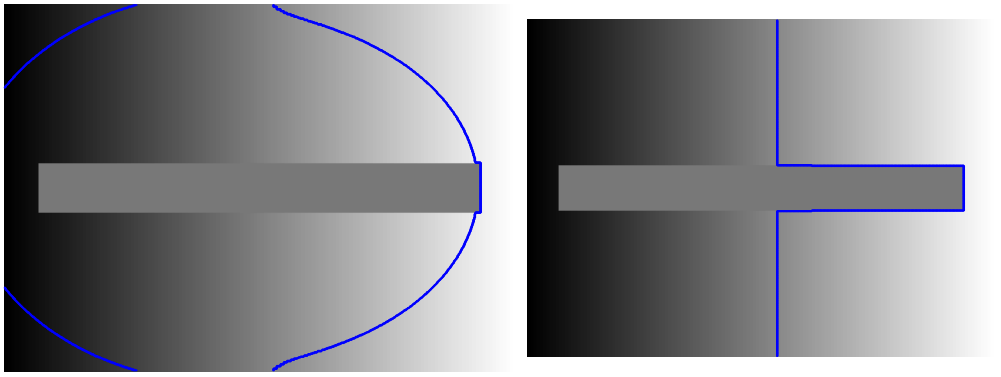


Figure 22 The segmentation of the gradient optical illusion. The image on the left shows the first iteration and the image on the right shows the sixth iteration.

our brain outperforms current computational algorithms when performing the same tasks. Thereafter, we choose one optical illusion in particular and analyse how the *active contours* performs the segmentation task. The optical illusion used is an example of a gradient illusion (see Fig.22). All the pixels in the central bar have the same intensity. This can be checked if the background is removed, isolating the bar. However, we perceive a bar with a gradient of intensity. The illusion results from the background gradient effect on the process of defining the reflectance of the object to which we are looking at (which in this case is the bar).

Our perceptual system has the ability to adjust to different light conditions, that is, it sees objects as continuing to have the same brightness even though light may change their immediate sensory properties. An object will exhibit *brightness constancy* as long as both the object and its surroundings are in light of the same intensity. If the background brightness differs from the object, brightness constancy is not maintained. For instance, a sheet of white paper seen in the bright sunlight reflects a very different amount of light than the same sheet of paper seen later that night in a softly lighted room. Yet we perceive the paper as having the same whiteness in each case. The gradient illusion breaks brightness constancy.

In Figure 22 we can observe that the segmentation method was not successful. This method uses the global properties of the image in order to achieve the energy minimization. If we consider a symmetrical division of the image along the y axis, the intensity mean for each domain is equal so that it is not possible to define different domains. In the case of Figure 21 (on the right) were defined two domains due to a break of symmetry. By repeating application of the method, it should be obtained, for half of the times, the result in Figure 21 (on the right) and the symmetric result (the left part of the bar) for the other half of the times.

2.14 Method to determine the pupil centre of mass

As we mentioned in the end of section 2.12, we used the information given by the segmentation method and developed a new and simple algorithm to locate the coordinates of the pupil centre. In spite of this approach, we did not

achieve the desired robustness, as it will be discussed. Overall, we established a new method to determine the pupil centre which does not uses the *active contour* approach.

We will first present a simple 1D example in order to explain clearly what it the framework of our method. Consider two functions $f(x)$ and $G(x)$, where $G(x)$ is a Gaussian function centred around an arbitrary position, X, and $f(x)$ is a square well (for the sake of simplicity) (see Fig.23):

$$f(x)=\begin{cases}0, x_0 \le x \le x_1\\ X, x_2 \langle x \le x_3\\ 0, x_3 \langle x \le x_4\end{cases} \qquad G(x)=\exp(-\frac{(x-X)^2}{2\sigma^2}) \qquad \textbf{(42)}$$

We want to find the coordinates around which the Gaussian should be centred so that the two functions have maximal overlap, i.e., their sum is minimum (see Fig.23). Hence we define an energy function given by $E(x)=\int_{x_0}^{x_4}(w(x))^2 dx$ where $w=f(x)+G(x)$, and (see Fig.23). The minimum of E is achieved when:

$$\begin{aligned}&\frac{dE}{dx}=\frac{\partial E}{\partial w}\frac{\partial w}{\partial x}=0 \Leftrightarrow\\ &\frac{dE}{dx}=2w(x)\left[\frac{\partial w}{\partial f}\frac{\partial f}{\partial x}+\frac{\partial w}{\partial G}\frac{\partial G}{\partial x}\right]=0 \Leftrightarrow \qquad \textbf{(43)}\\ &\frac{dE}{dx}=2(f(x)+G(x))\left[0-\frac{(x-X)}{\sigma^2}G(x)\right]=2(f(x)+G(x))\left[\frac{(x-X)}{\sigma^2}G(x)\right]=0\end{aligned}$$

A discretization of these equations leads to:

$$2\sum_i (f(i)+G(i))\left[\frac{(i-I)}{\sigma^2}G(i)\right]=0 \quad \textbf{(44)}$$

Thereafter, the minimum can be calculated iteratively using the fixed point iteration method, as we did in Eq.31:

$$I^{n+1}=I^n \pm kT(I^n) \quad \textbf{(45)}$$

where:

$$T(I^n)=\sum_i (f(i)-G(i))\left[\frac{(i-I^n)}{\sigma^2}G(i)\right] \quad \textbf{(46)}$$

i^n is the value of i on the i[th] iteration and k is a constant that accelerates or decelerates the convergence. In this case we used $k = 0.001$.

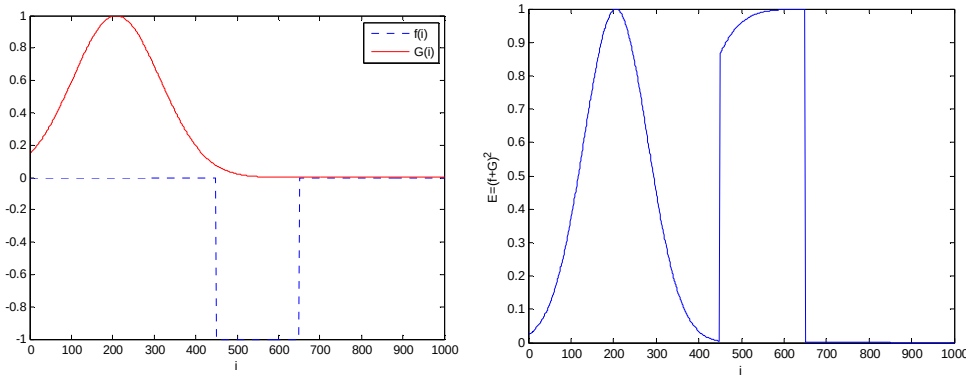


Figure 23A) First Iteration of the complementary method in the example. On the left functions $f(i)$ and $G(i)$ are represented. On the right we have the representation of the function $E(i)$. The initial position of the maximum of the Gaussian curve is 200.

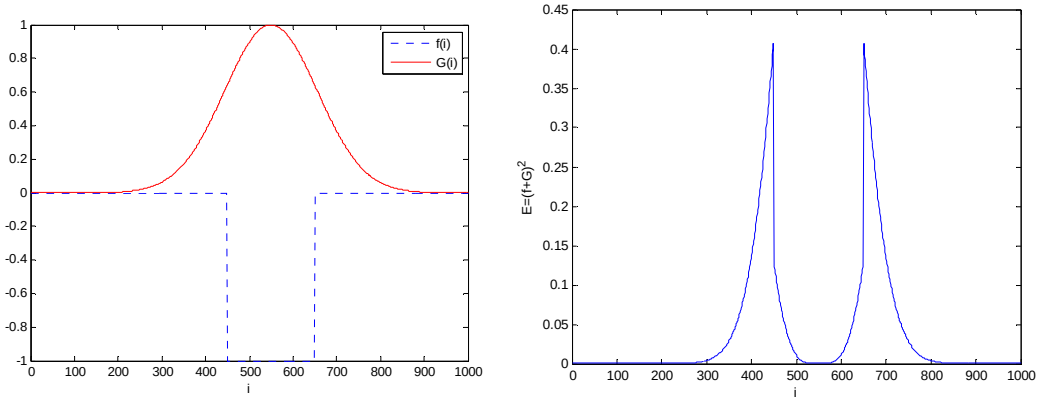


Figure 23B) Last iteration of the method (22). As we can see the Gaussian moved so that the minimum of $(f(i)+G(i))^2$ is achieved when the centre of the Gaussian is at $i = 550$.

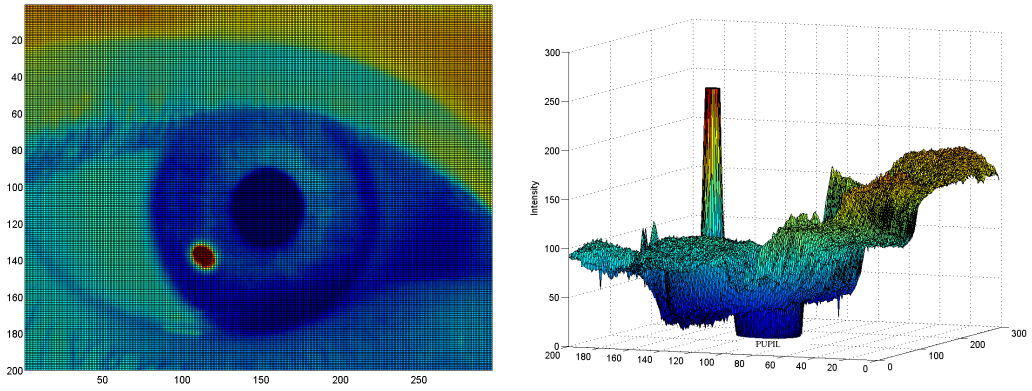
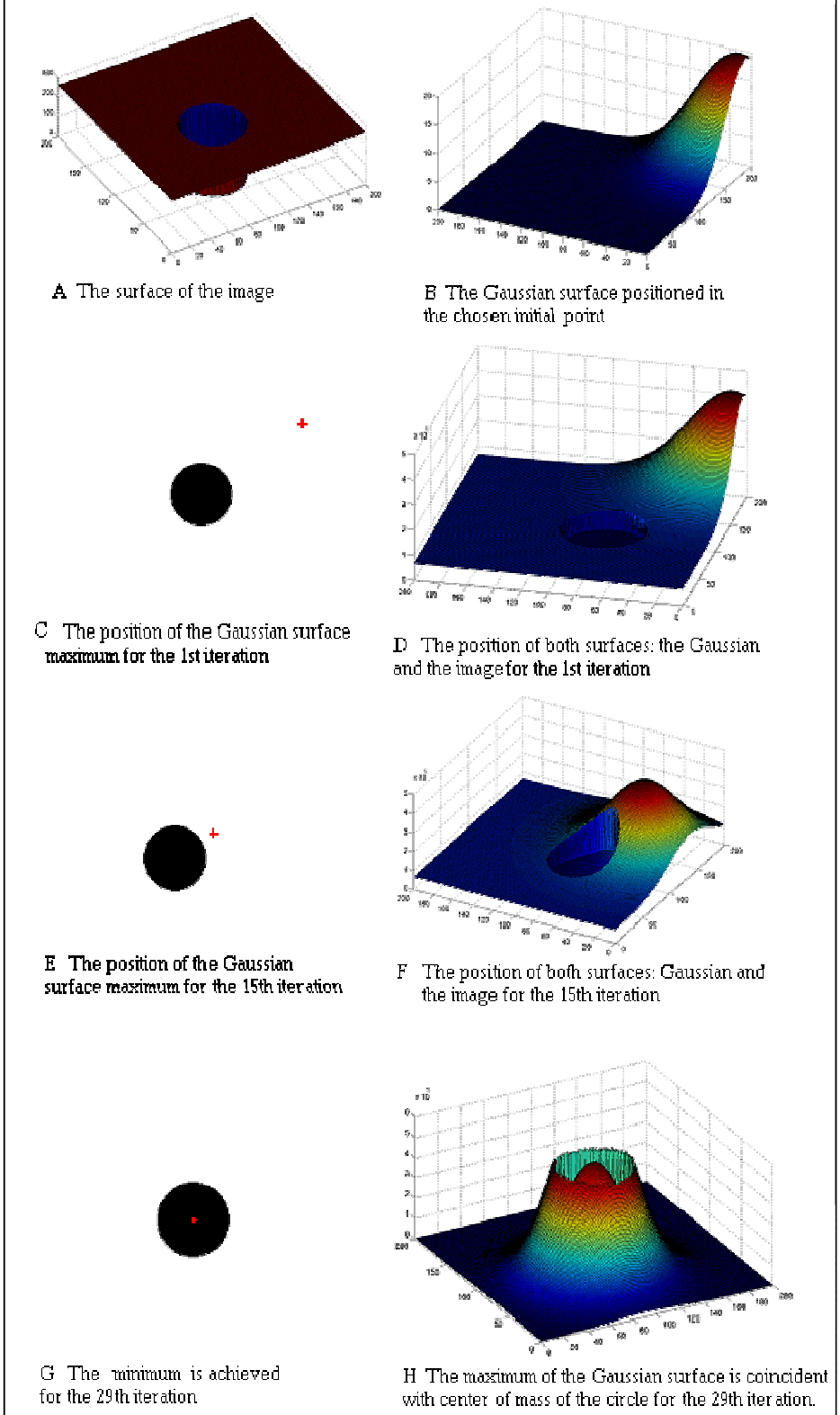


Figure 12C) On the left we can see the projection on planes x and y of the surface associated to a typical eye image obtained. On the right we can see the surface of pixel intensities. There are similarities between this surface and the $f(i)$ function from the previous example.

In this way the position of the maximum of the Gaussian curve moves along the time and the convergence is achieved when the overlapping is maximal

(minimum of $E(i)$ is achieved)) is achieved. An example can be observed in Figure 23A, B, C. Although this is a simple method, it is interesting to show it works. In the Figure 24 the determination of the centre of mass of a circle is shown. The Gaussian surface has a $\sigma = 40$ pixels so that 2σ is larger than the circle radius which equals 60 pixels. Images D, F and H show $(I + 20 \cdot G)^2$ where I is the image surface and G is the Gaussian surface. G



A The surface of the image

B The Gaussian surface positioned in the chosen initial point

C The position of the Gaussian surface maximum for the 1st iteration

D The position of both surfaces: the Gaussian and the image for the 1st iteration

E The position of the Gaussian surface maximum for the 15th iteration

F The position of both surfaces: Gaussian and the image for the 15th iteration

G The minimum is achieved for the 29th iteration

H The maximum of the Gaussian surface is coincident with center of mass of the circle for the 29th iteration.

Figure 24 Application of the method to a circle.

has a factor of 20 so that its possible to observe the Gaussian maximum in the figure . For calculations this factor was not considered.

In eyes' images, there is more than one dark object (see Fig.20A). Hence, the method was also applied to more complex examples, similar to the example in Figure 20A. What was observed was that the Gaussian surface would not always be centred in the centre of mass of the pupil depending on:

- the initial coordinates of the centre of the Gaussian;
- the sigma of the Gaussian surface;

Figure 25 shows some typical examples.

If we look to the Figures25M and 25P, we can see that the intermediate positions are close to the centre of mass of the circle. In spite of this the

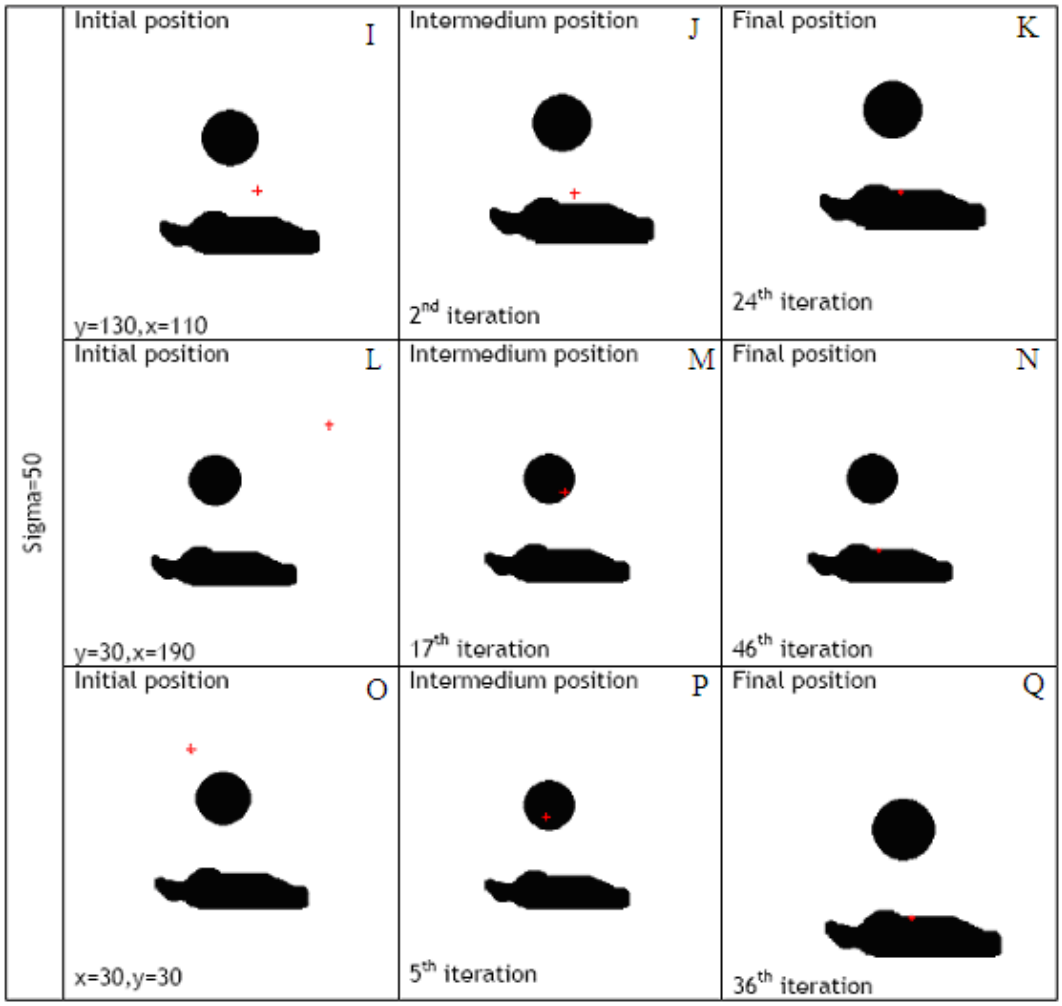


Figure 25 In this example we can observe that different initial points can have different convergence points.

method does not converge to the centre. The value of sigma is too large (larger than the radius of the circle) the second object competes strongly by the position of minimum. We can conclude that the centre of mass of the circle was not determined correctly. This happened because the value of sigma was too large allowing the second object to contribute largely to the minimum of energy.

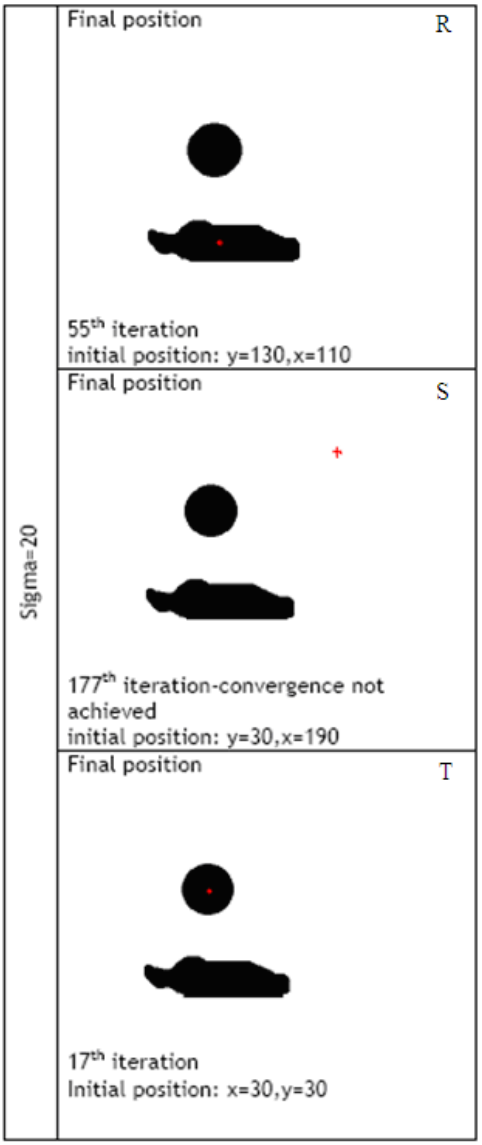


Figure 26 The final positions associated with a sigma equal to 20.

In Figure 27 are shown the examples of the results of the application of the method, using the initial points previously considered (see Fig.25), but a smaller sigma, $\sigma = 20$. In Figure 26R the minimum converged to a position inside the second object. The contribution of the circle decreased so that the major contributions were from the second object. In Figure 26S, the maximum was not achieved because there were no contributions from both objects as the sigma was insufficiently large. In Figure 26T, we can see that the method converged to the desired position. This was a result of a compromise between the initial point which was sufficiently close to the circle and sufficiently far from the second object. In this case σ that was not sufficiently large to consider the contributions from the second object.

Since the method was not robust as it depended on the initial conditions, it was necessary to have an estimation of the centre of mass of the pupil, so that the initial conditions could be wisely chosen. After the application of the active contour without edges, we defined a way of guaranteeing the compromise between the parameters involved in the complementary method: the value of sigma and the position of the initial point. In order to choose an appropriate initial point for the fixed iteration point method, we defined that this point should be the centre of mass of the corneal reflection. We also defined a region of interest in the image in order exclude the possibility of competition between dark objects. Furthermore, as the number of elements in the matrix decreased, the computational time improved considerably. We think this method is more robust than the *active contours without edges* for the case of our images.

In Figure 27 we can observe the application of the method. The initial point of start for the method is the corneal reflection centre of mass (white spot). Using this point and the value of sigma, a region of interest is defined. As there is a relation between the position of corneal reflection and the position of the pupil, the problems associated with fast variations of the position of the pupil are minimized. In Figure 27A is represented the position achieved in the second iteration. In Figure 27B is represented the final position achieved with the method after 21 iterations. This corresponded to a computational time equal to 0.841 seconds. As we can see the shape of the pupil has not eccentricity exactly equal to the eccentricity of a circle. This means that using the Gaussian to determine the pupil centre is an approximation. In order to minimize the error associated to this approximation, we considered an additional procedure. When the convergence is achieved ($I^{n+1} - I^n < 0.001$), the value of I^{n+1} is disturbed for a small value (see Table 1) and the method is applied again. However the precision of the eye tracker is not affected. For tasks which goal is not to determine its accurate position but to assess the variation of the movements of the eye, this aspect of the method is not determinant.

Point of convergence	Disturbance x=15 Disturbance y=15	Disturbance x=-15 Disturbance y=15	Disturbance x=15 Disturbance y=-15	Disturbance x=-15 Disturbance y=-15
41.2558 44.4947	41.2582 44.4947	41.2579 44.4947	41.2582 44.4944	41.2578 44.4945

Table 1 Typical results after achieving the point of convergence. We summed disturbances along x and y and performed the method again. The results gave variations of the order of the 0.001. Hence, this not solve the problem associated with the eccentricity of the pupil

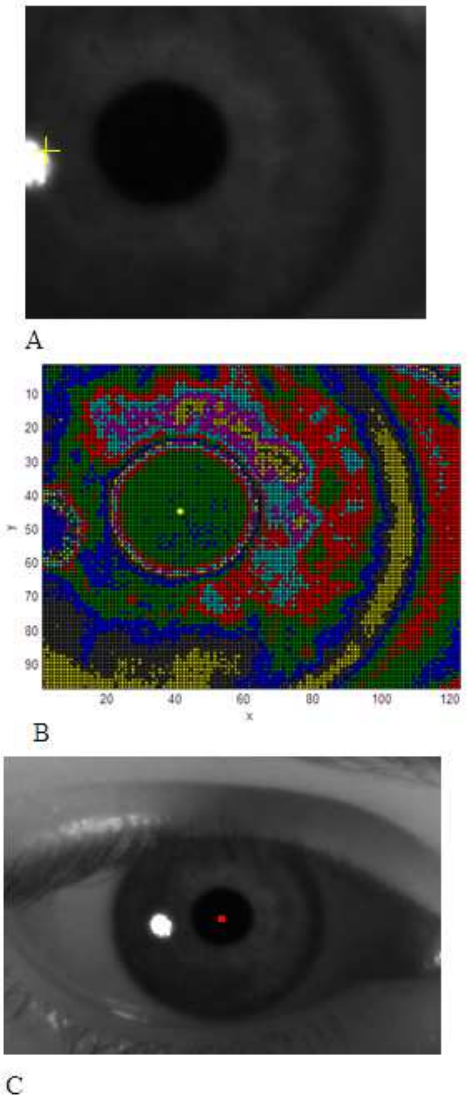


Figure 27 In A are represented the coordinates obtained by the method in the second iteration, inside the region of interest (97pixelsx123 pixels). In B is a graph that shows the centre of mass obtained by the method (yellow dot) after 21 iterations. This graph was obtained using the MATLAB instruction colormap (lines);pcolor(I). The colours are associated with intensity values. This representation was used in order to show a typical shape obtained for the pupil. In C it is represented the position of the pupil centre obtained. The image had a 200pixelsx200pixels resolution so using the region of interest represents a reduction of 70% in the size of the matrix.

3. NEURAL NETWORKS

3.1 Introduction

Neural Networks are a mathematical framework that aims to reproduce the same type of intelligent behaviour as that displayed by humans. Its architecture was inspired in the human brain, and consequently their constituents are also called neurons and their interactions are called synapses. These serve to receive and send information. The information that reaches a neuron can arrive from different synapses. These can have different impact on the neural response. Hence, each neuron can integrate signals coming from different channels simultaneously, and also send information to several neurons at the same time. The importance of each connection depends on the training that is given to the brain. For instance, while we are children we train for a long time how to grab an object properly. That process of trial and error influences the way connections are established between neurons. As a result, the brain gains the ability to:

- Recognize patterns in the presence of noise;
- Recall memories;
- Make decisions for current problems based on prior experience;

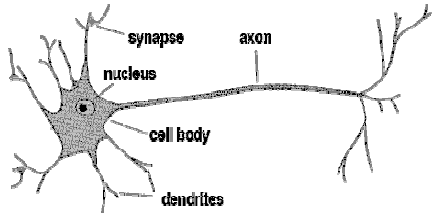


Figure 28A) The biological model of a neuron.

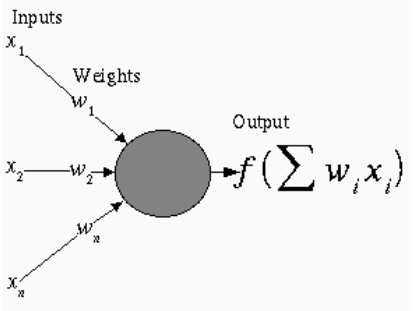


Figure 28B) The mathematical model for a neuron. Neurodes in an Artificial Neural Network receive signals from other neurodes or from external sources, perform transformations on them, and then pass those signals on to other neurodes.

As can be observed in Figure28, Neural Networks intend to reproduce these capabilities in a more basic way. The biological model of the neuron also receives information, activating the neuron and producing an output. Also, each neuron is connected to other neurons (synapses).

There are many different types of neural networks depending on the task they are planned for. Each type of neural network is specified by a neuron model, a network architecture and a learning algorithm. The neuron model is defined by a response function which establishes how an output corresponds to a certain input, that is, activates the neuron. The architecture establishes the number of neurons that compose the network. The learning algorithm defines the process through which the network increases the intensity of certain interactions (i.e., their weights) and not others. This is essential to establish the correct association between inputs and outputs. The input information and the connection (synaptic) weights are the arguments of the

transfer function that establishes the response of the neuron, i.e., its output. During a learning process, the output is compared with a target which establishes what output should have been associated to a certain input. Depending on a comparison between the target and the actual output becomes closer to the target. This process is designated the training of the network. After the training, we can say that the network is prepared to output new information about the system, even for inputs that have never been presented to the network during the learning process. This ability of predicting targets to new inputs, is called the generalization. This opens the possibility to deal with complex problems in a relatively simpler way. This explains the importance of Neural Networks in a broad range of applications (e.g. classification problems, mathematical modelling, forecasting, clustering, parallel processing, hardware design, cognitive learning, function approximation, etc.).

3.2 Neuron Model

The neuron consists of:

- a set of links with weights $w_1, w_2, w_3, ..., w_m$,
- a summation function that computes the weighted sum of the inputs $\sum wx$,
- a transfer function that evaluates the strength of the input.

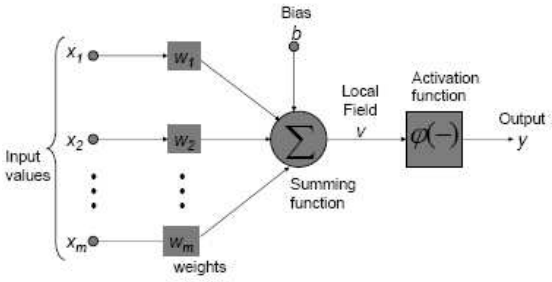


Figure 29 In this model we can observe the set of inputs and links, the summation function and the transfer function.

Therefore, the output is determined by $y = \varphi(\sum wx)$. Several functions can be used as transfer functions φ, as shown, in Figure 30. Several kinds of transfer functions

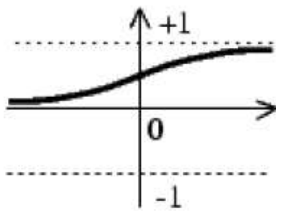


Figure 30A Log-sigmoid function

$$\varphi(t;a,m,n,\tau) = a\frac{1+m\cdot\exp(-\frac{t}{\tau})}{1+n\cdot\exp(-\frac{t}{\tau})}$$

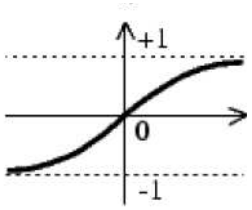


Figure 30B Tan-sigmoid

$$\varphi(t) = \frac{2}{\sqrt{\pi}}\int_0^t \exp(-t^2)dt$$

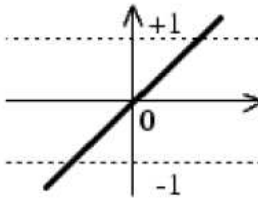


Figure 13C linear function

$$\varphi(t,a,m) = m\cdot t + a$$

can be used; the only requirement is that they are differentiable, so that the back-propagation algorithm can be derived.

For our work, we used a linear function for the hidden layer and a linear function for the output layer. As we used the network to approximate a

function, the output of the network should be a value not subjected to constraints, that is, $\varphi \in \Re$.

3.4 Architecture

In a neural network, the network can be formed by several neurons and several neuronal layers. A network has to have at least two layers: the input and the output ones. The neurons between those two layers are hidden layers as they are not directly connected with inputs or outputs. They are designated as hidden layers. The information can be fed into the neurons forwardly or recurrently. For our work we used the first kind of feed.

3.5 Learning Algorithm

There are two main groups of learning algorithms: supervised and unsupervised. The former is based on minimization processes and the last relies on autocorrelation relations. Here we will only study a supervised learning algorithm. A supervised learning algorithm is divided in four steps:

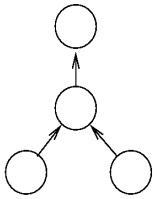


Figure 31A This is an example of a multilayer feedforward network. Each neuron output is the input for the next layer of neurons.

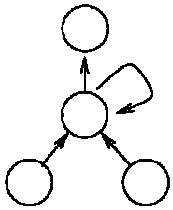


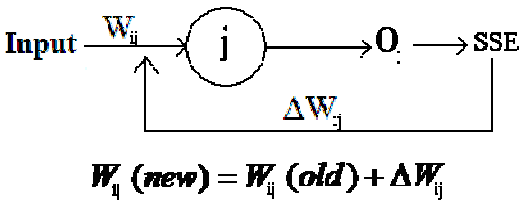
Figure 31B This is an example of a multilayer recurrent network. Not only each neuron output is the input for the next layer of neurons but also a neuron output can be fed into that same neuron.

- Determination of the summation function;
- Calculation of the outputs for every neuron;
- Calculation of the errors for the outputs;
- Calculation and apply the weight adjustments using the errors of the outputs;

For our work, the weight adjustments were calculated using the Back-propagation algorithm.

3.6 Back-propagation

The Back-propagation algorithm minimizes the difference between the desired values-targets- and the outputs of the network. The output errors for each node depend on the weights of the inputs of that node. In order to minimize the errors, adjustments of the weights must be done. This algorithm establishes that new



$$W_{ij}\,(new) = W_{ij}\,(old) + \Delta W_{ij}$$

Figure 32 The generalized delta rule determines the adjustment of the weights, back-propagating the errors.

weights depend on the old outputs so that there is a back-propagation of the output errors to the weights adjustments (see Fig. 32). Mainly, we are looking for the relation that evaluates the errors variation as function of the weight's variations:

$$\frac{\partial SSE}{\partial W_{ij}} = \frac{\partial SSE}{\partial O_j}\frac{\partial O_j}{\partial I_j}\frac{\partial I_j}{\partial W_{ij}} \quad \textbf{(47)}$$

Here SSE is the sum of the squared errors associated to the weight W_{ij}, $SSE = \sum_j (D_j - O_j)^2$; O_j is the output of neuron j, where $O_j = \varphi(I_j)$ and where r D_j is the target to neuron j. Finally, I_j is the net weighted input received by neuron j, that is, $I_j = \sum_{j=f}^{s} W_{ij} O_i$. Given that, $\frac{\partial SSE}{\partial O_j} = -2(D_j - O_j)$ and $\frac{\partial O_j}{\partial I_j} = \frac{\partial O_j}{\partial \varphi_j}\frac{\partial \varphi_j}{\partial I_j}$ and $\frac{\partial I_j}{\partial W_{ij}} = O_i$.

The back-propagation method uses gradient descent learning, that is, it minimizes the error by moving weights along the slope of error. This movement follows the direction that yields the maximum decrease of the network error. That direction is the opposite of the gradient of SSE, that is, there is a relation of proportionality between ΔW_{ij} and $\frac{\partial SSE}{\partial W_{ij}}$:

$$\Delta W_{ij} = -\eta \cdot \frac{\partial SSE}{\partial W_{ij}} \quad \textbf{(48)}$$

The generalized delta rule defined by $W_{ij}(new) = W_{ij}(old) + \Delta W_{ij}(new)$ (Rumelhart *et al*, 1986) where:

$$\Delta W_{ij} = -\eta \cdot \delta_j O_i \quad \textbf{(49)}$$

where $\delta_j = \frac{\partial SSE}{\partial I_j} = \frac{\partial SSE}{\partial O_j}\frac{\partial O_j}{\partial I_j}$ is the local gradient of neuron j and $O_i = \frac{\partial I_j}{\partial W_{ij}}$ is the output of neuron i. We used in our network architecture two kinds of transfer functions: linear function where $\frac{\partial O_j}{\partial I_j} = m$ and tan-sigmoid function for $\frac{\partial O_j}{\partial I_j} = \frac{2}{\sqrt{\pi}}\exp(-I_j)^2$. Hence, the local gradient of neuron j is $\delta_j = -2m \cdot O_i \cdot e_j$ for

the linear function or $\delta_j = -2O_i \cdot e_j \frac{2}{\sqrt{\pi}} \exp(-I_j)$ for the tan-sigmoid function. The extended version of the generalized delta rule is given by:

$$\Delta W_{ij}(new) = \eta \delta_{ij} O_j + \alpha \Delta W_{ij}(old) \quad \textbf{(50)}$$

where η is the learning coefficient and α is the momentum coefficient. η is a constant that introduces in $\Delta W_{ij}(new)$ a fraction of the influence of several most recent observations so that the error surface becomes smoother The learning coefficient is a value major than 0 and lesser or equal than 1 which guarantees that the change in weights, $\Delta W_{ij}(new)$, is not to slow or to rapid. If η is set too low, many epochs are required to train the network. If η is set too high, the weights may oscillate wildly about some optimal value, originating instability in the training process. The momentum coefficient is like a filter for the error surface. Often, in optimization problems local minima can interfere in the process of achieving the global minima. A local minimum may induce small variations of the weights as if the network is reaching a global minimum. Therefore, α influences the minimization process to produce larger variations of weights despite the tendency to reduce them. If this coefficient is too high it may induce large variations of weights even when it achieves the global minimum. The learning coefficient and the momentum coefficient accelerate or decelerate weight changes.

The local gradient was defined for a neuron located in the output layer of the neuron. For a neuron located in a hidden layer the desired output is not defined. Hence, the error for the hidden neuron i would have to be determined recursively in terms of the output error of the neurons to which neuron i is connected. In order to calculate the local gradient for the hidden neuron i we have to calculate the following partial derivative

$$\frac{\partial SSE}{\partial O_i} = \frac{\partial SSE}{\partial e_j} \frac{\partial e_j}{\partial O_i} = \sum_j e_j \frac{\partial e_j}{\partial O_i} \quad \textbf{(51)}$$

Using the chain rule we will have $\frac{\partial e_j}{\partial O_i} = \frac{\partial e_j}{\partial I_j} \frac{\partial I_j}{\partial O_i}$ where $\frac{\partial e_j}{\partial I_j} = \frac{\partial \left(D_j - \varphi_j(I_i)\right)}{\partial I_j} = -\varphi_j(I_i)$ and $\frac{\partial I_j}{\partial O_i} = W_{ij}$. Thereafter, Eq. 51 becomes $\frac{\partial SSE}{\partial O_i} = -\sum_j \delta_j W_{ij}$. The back-propagation formula for the local gradient for a neuron in a hidden layer is $\delta_j = \varphi_j(I_j) \sum_j \delta_j W_{ij}$.

In the next section we will discuss how we used neural networks in our work.

4. THE EYE TRACKER

4.1 The camera

The camera used for acquiring images is a PixeLink camera, A-741. This camera can acquires images at frame rates up to 300 frames per second for resolutions of 296×200. Its spectral response is between $350 - 1000nm$ so it is sensible to infra red light. This is very important as the eye will be illuminated with an infra red lamp. It is possible to adjust several parameters during the acquisition such as the exposure time (which is determinant to define the frame rate), the gain (which can increase the luminance in the image but can also introduce noise) and the gamma factor (which can increase contrast).

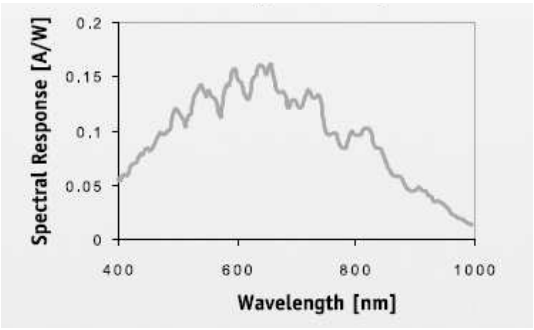


Figure 33 The spectral response of the camera

The PixeLink A-741 software uses C language functions which can be used to build a specific and personal program. It was observed that the acquisition of images with very high frame rates with the software included with the camera (PixeLink Capture OEM) could lead to instability of the computer system. This problem does not occur using the built in C language functions in a C environment (e.g. Visual C++). The possibility of building specific programs is very important if there is the need to implement an experimental procedure that requires synchronism between the acquisition and the task the subject is performing. For instance, if there is some information that is in the screen and the subject has to acquire that information and then perform the task, it is useful to acquire image of the movement of the eyes only when the subject starts the task.

4.2 The lens

A lens was chosen for the camera considering the kind of flexibility that we think the eye tracker must have. Hence, we estimated that the subject would be at distance $D = 1000mm$ away from the camera.

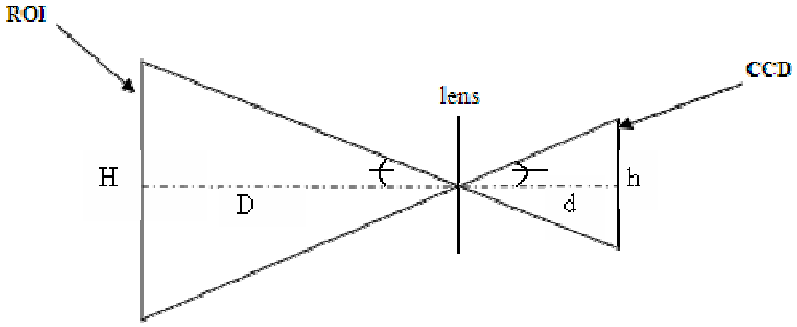


Figure 34 The lens has to allow a ROI approximate to the area of one eye so its focal length, *f*, should be equal to *d*.

The CCD of the camera has a resolution of $640\,pixels \times 480\,pixels$. This corresponds to an area of $4.288mm \times 6.7mm$. H was estimated to be $40mm$ so that the area corresponding to the eye would be $40mm \times 40mm$. Using the angular relation represented in Figure 34 we will have:

$$\frac{H}{2D} = \frac{h}{2f} \Leftrightarrow f = \frac{h}{H} D = \frac{4{,}3mm}{40mm} \cdot 1000mm \approx 100mm \quad \textbf{(52)}$$

which was used to choose the lens that would be more adequate for our setting.

4.3 The Infra Red Lamp

The infra red lamp is composed by several infra red LEDs, with wavelengths of $850nm$. Although it is not the wavelength to which the camera is most sensible it is sufficient for a good illumination. This lamp is used in order to achieve the desired exposure time ($5-3ms$) which consequently lead to $200-300$ frames per second.

The lamp was positioned $10cm$ from the eye. The risk of damage to the eye associated to infra red light is high if there is heat associated. Since this lamp is constituted by LEDs it has a low power density, reducing this risk.

The corneal reflection (CR) is a reference since relation between its position and the pupil's position remains constant during head translation, but moves with eye rotation.

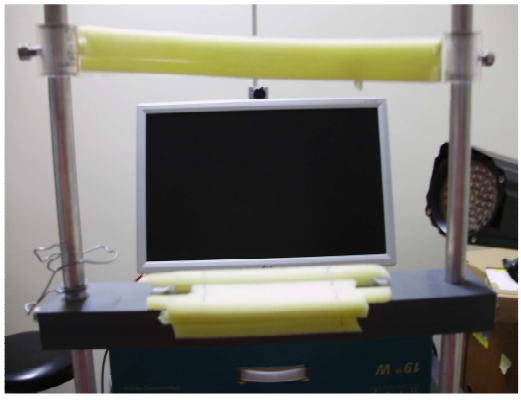


Figure 35 The subject's perspective.

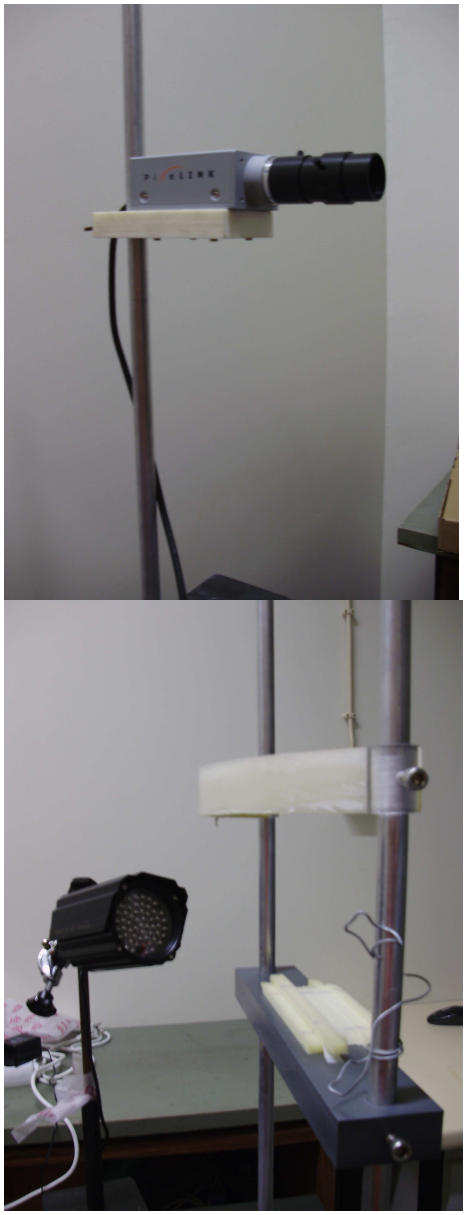


Figure 36 The camera (upper image), head fixed system and the infra red lamp (lower image).

4.4 The experimental set up

The eye tracker we developed is a head fixed one, that is, the subject must seat with the head still in order to eliminate other movements than the movements of the eyes. The screen has a resolution of 1162x862 pixels. The camera was set to point towards the right eye.

4.5 Calibration using Neural Networks

The calibration of the eye tracker consists in establishing the relation between the CR position, the pupil position and the point of gaze.

The experimental procedure followed was: the subject was asked to look to points on a screen, presented one at a time. The staring lasted approximately 5 seconds for each point (to stabilize the gaze and take a photograph). Two different sets of points were presented to the subject where each set was presented from left to right and from the bottom to top.

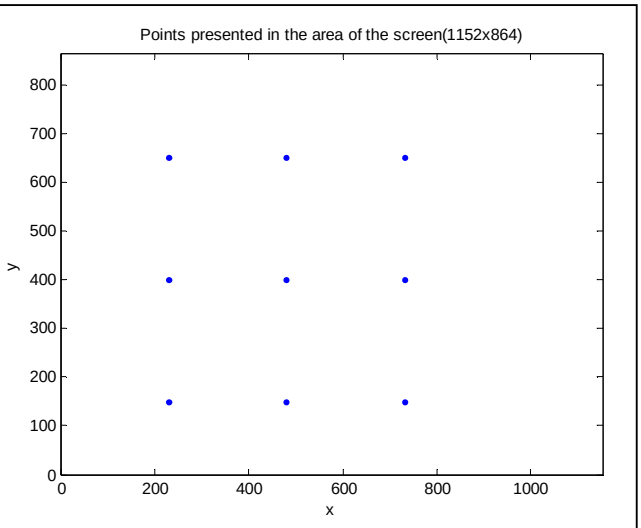


Figure 37 The grid of points presented in the screen, one at a time.

In the first set of experiments, nine points equally spaced (89mm for x coordinate and 74mm for y coordinate) were presented to the subject, repeatedly for three times to collect information from the same point of gaze (see Fig.27). In Figure 38 are represented the trajectories of the pupil centre of mass and the CR centre of mass in the domain of the eye image (296pixelsx200pixels), associated with this set. As we can see there is a defined structure for both trajectories. This structure is similar to one associated to the set of points presented. This shows that the eye follows a defined trajectory while looking to each presented point. This observation is confirmed for the graphs in Figures 39 and 40 where

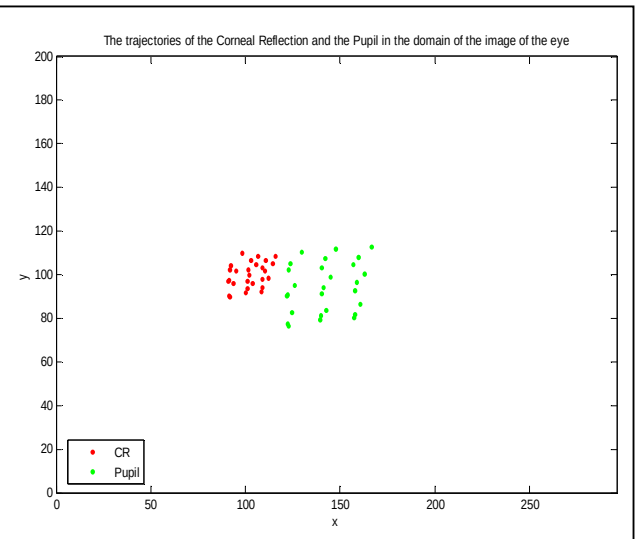


Figure 38 The trajectories of the Pupil and the CR in the domain of the eye image.

we can observe the differences of positions between the pupil and the CR, along both coordinates. As were performed three repetitions are represented three sets of data: data1 (red), data2 (green) and data3 (blue). The differences along x and along y remain approximately constant for all three sets. In Figure 40 we can also observe that the differences along y, for each three positions on the screen corresponding to points with the same value for y, remain almost constant.

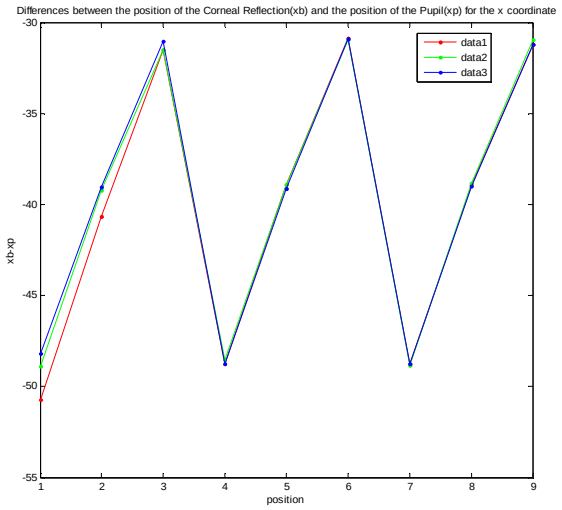


Figure 39 Differences between the pupil and CR positions, along x.

In the second set of experiments, twenty five points equally spaced (45mm for x and 37mm for y) were presented to the subject. These points had smaller periodicity between them in order to better assess the eye tracker precision. In Figure 41 (left) is represented the trajectories of the pupil and the CR. Comparing these results with the obtained with the previous set, we can see that for the pupil trajectory the space between each pupil position has decreased which also happened to the CR position. These results can be better observed in Figure 42, where the distances between the pupil and the CR have decreased. These results confirm that there is a relation between the pupil, the CR position and the set of points presented in the screen (establishes the point of gaze). Because the differences vary for the different sets of points presented and remain approximately constant for equally spaced points presented. In Figure 44 is represented the scatter plot of the differences along x and y. Comparing this figure with Figure 41(left) we can see that the values in Figure 44 correspond to a transformation of the values in Figure 41(left). That transformation does not have the same effect for each value it is applied. For instance, the values with smaller differences along x and the ones with larger differences along x, show different behaviour. As this kind of effect are not associated with linear transformations we decided to use a neural network to

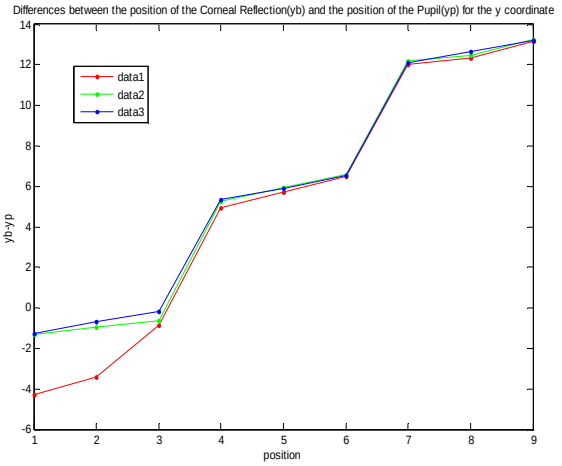


Figure 40 Differences between the pupil and CR positions, along y.

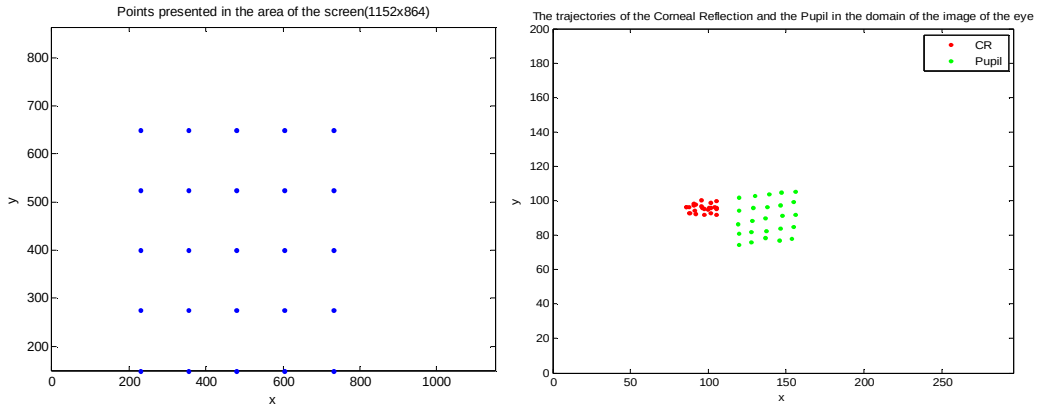


Figure 41 On the left, is the grid of points presented to the subject, one at a time. On the right, are the trajectories of the pupil and the CR in the domain of the eye image.

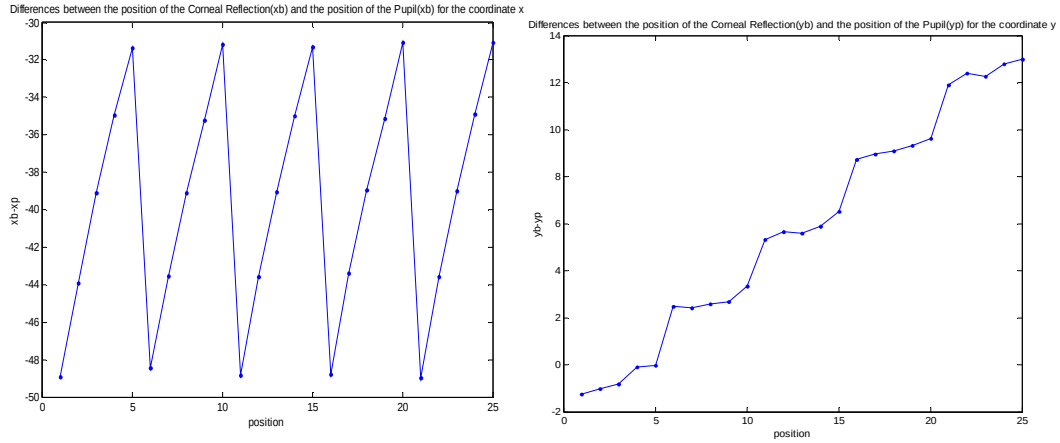


Figure 42 Differences between the position of the Pupil and the position of the CR, for x (on the left) and y coordinate (on the right).

approximate the function that establishes the relation between the position of pupil centre of mass, the position of CR centre of mass and the point of gaze.

Neural networks usually generalize well to unseen data. Hence, we trained a network and used it to approximate the calibration function. Back-propagation is one of the algorithms usually used to approximate functions. In order to have a network that generalizes well, there must be a trade-off between robustness and accuracy. The *universal approximation theorem* (Stone, 1948) insures that it is possible for a feedforward network, with sufficient number of units

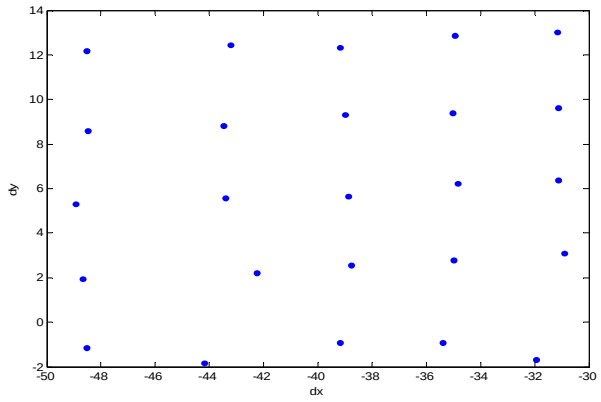


Figure 43 The scattering plot of the differences along x and the differences along y.

and trained using back-propagation, to approximate any continuous function to any level of accuracy. While building a network, one of the most important aspects to consider is the number of hidden units. An excess of units can lead to over-fitting, that is, the network becomes so complex that it is not flexible to generalization. Other important aspects are the transfer function and the number of epochs that we use to train the network. An excessive number of neurons can lead to worse prediction ability, that is, overtraining. In general, it can be a complex problem to choose a proper architecture.

In order to gain a better understanding about neural networks, we defined a similar function approximation problem and discussed different architectures . All of our work in Neural Networks was developed using the MATLAB Neural Networks Tool Box. In Figure 44A are represented the training set and target. The simulated data was given by the next equation:

$$\begin{bmatrix} w \\ z \end{bmatrix} = \begin{bmatrix} 2 & 0.1 \\ 0.1 & -1 \end{bmatrix} \begin{bmatrix} x + randn(1) \\ y + randn(1) \end{bmatrix} + \begin{bmatrix} 10 \\ 20 \end{bmatrix} \quad \textbf{(53)}$$

where $\begin{bmatrix} x + randn(1) \\ y + randn(1) \end{bmatrix}$ is the input with Gaussian noise and the $\begin{bmatrix} w \\ z \end{bmatrix}$ is the target. The training set was a matrix with 2×54 elements. These values of x and y were equally spaced by 100 units. Then, were built two other matrices: one with the same dimension and periodicity and the second with a smaller periodicity (25 units) and higher dimension (2×175) (see Fig. 44B and C). This last matrix is to assess the ability of generalization. Because it will require that

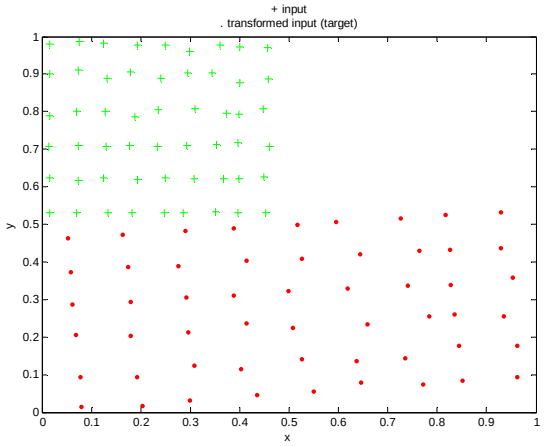


Figure 44A) The training set and the target.

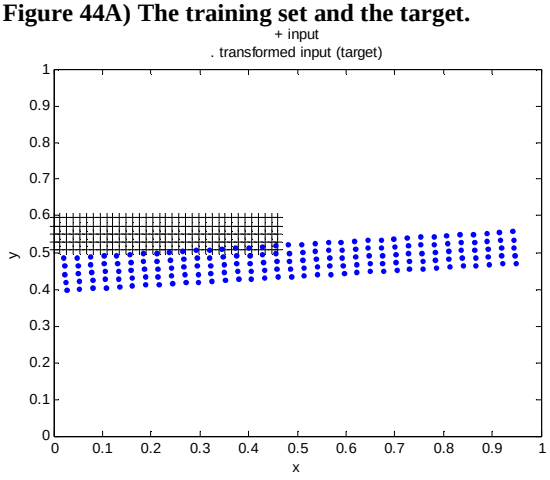


Figure 44B) The validation set and the target.

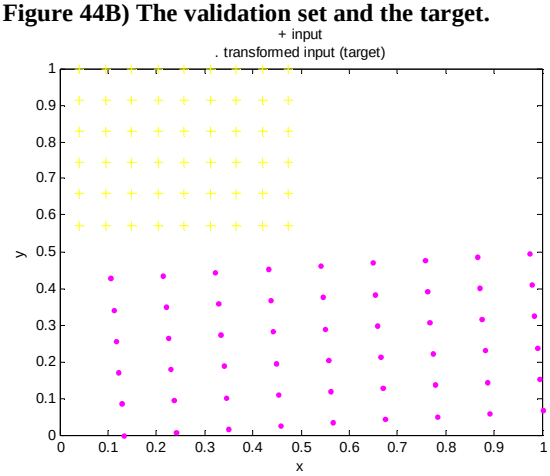


Figure 44C) The validation set and the target.

the function defined by the network to output extrapolated and interpolated

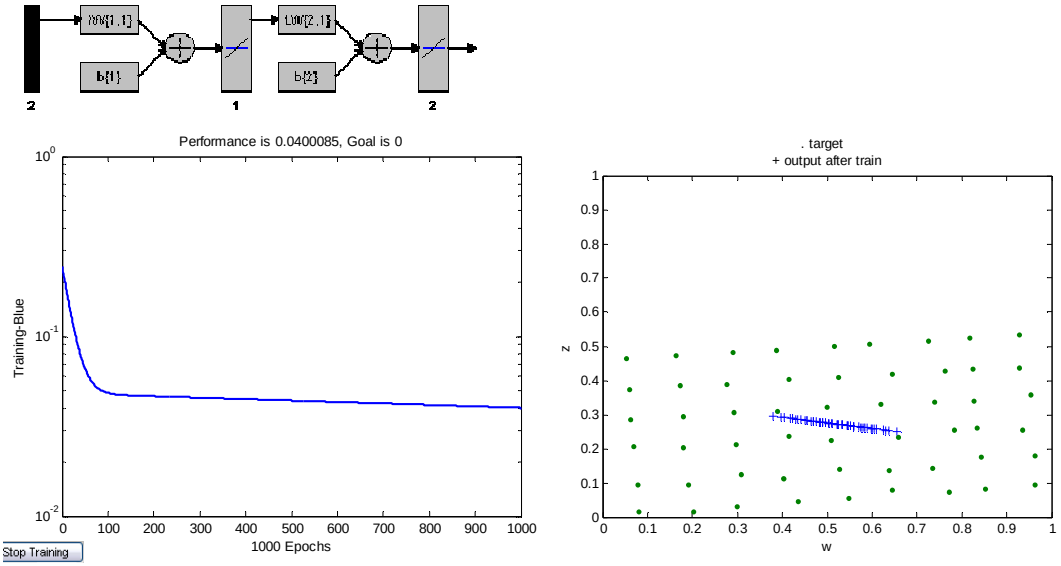


Figure 45A) On the left is represented the convergence of the network for 1000 epochs. On the right is represented the approximating done by the network to the target.

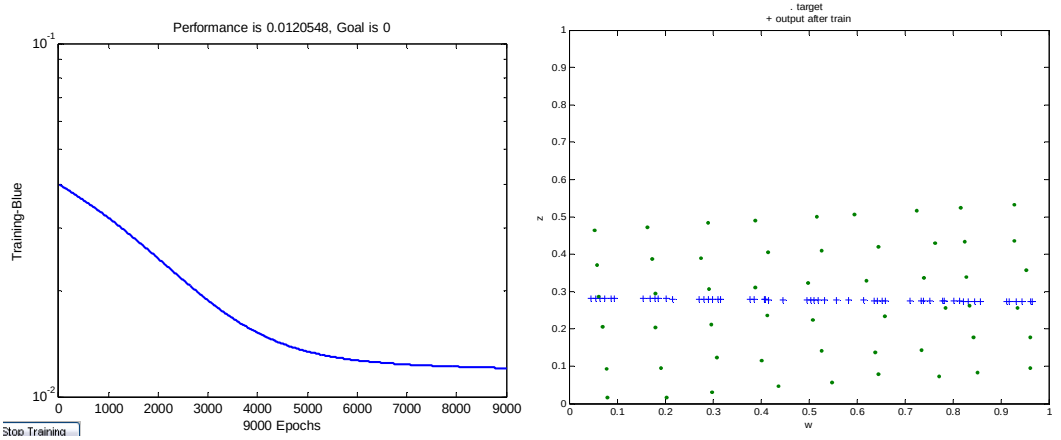


Figure 45B) On the left is represented the convergence of the network for 9000+1000 epochs. On the right is represented the approximating done by the network to the target.

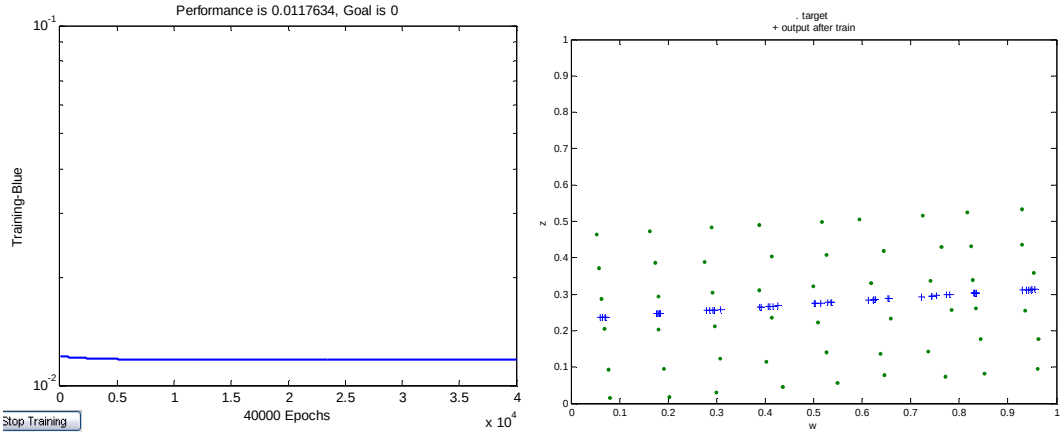


Figure 45C) On the left is represented the convergence of the network for 9000+1000+40000 epochs. On the right is represented the approximating done by the network to the target.

For the first network, we can see in Figure 45 the evolution of the mean squared errors has a function of the epochs. There is a higher decrease of the error for the first 10000 epochs and after 40000 the value only decreases by $0.0121 - 0.0118 = 0.0003$. The resultant approximation is very poor.

For the second network, we can see in Figure 46 that after 1000 epochs

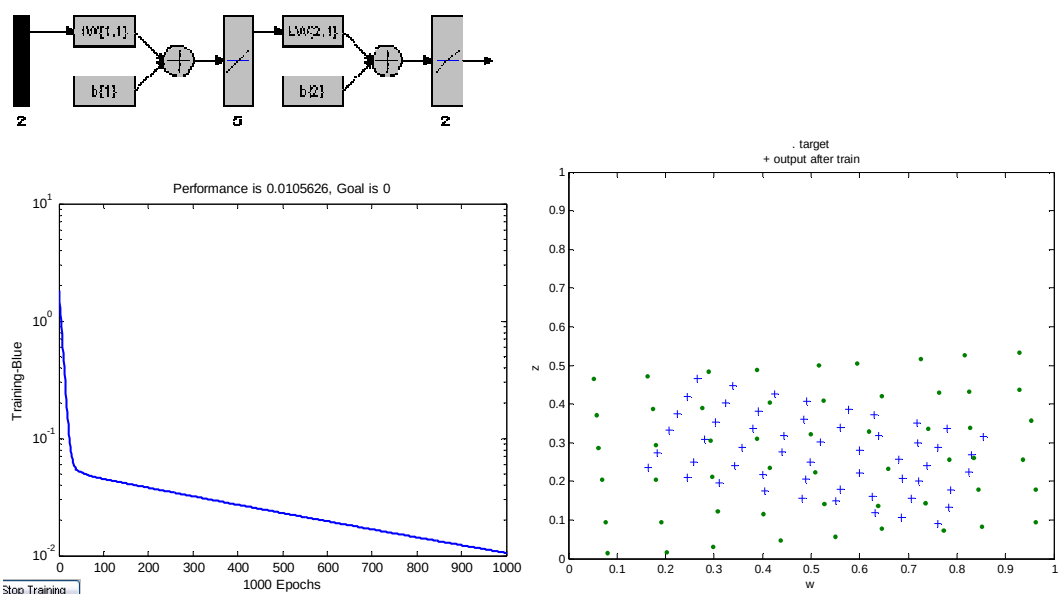


Figure 46A) On the left is represented the convergence of the network for 1000 epochs. On the right is represented the approximating done by the network to the target.

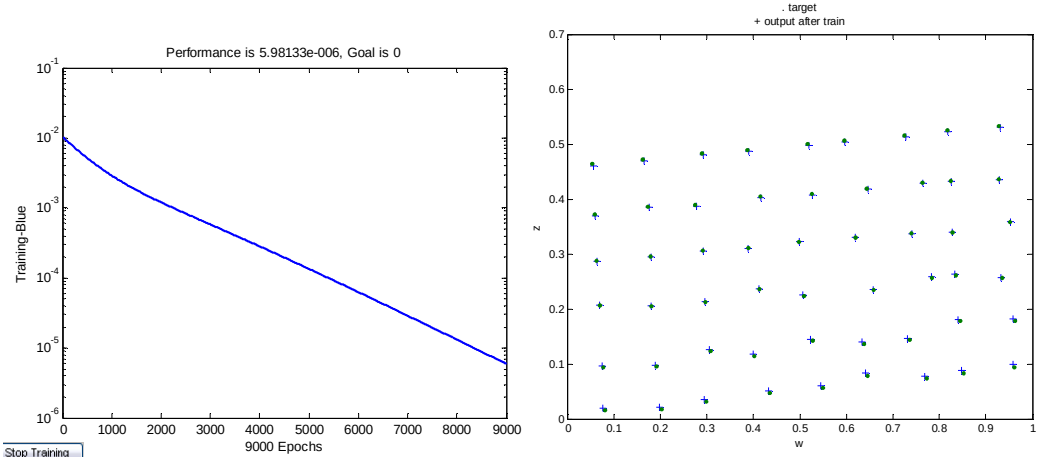


Figure 46B) On the left is represented the convergence of the network for 9000+1000 epochs. On the right is represented the approximating done by the network to the target.

the approximation is not good but after 10000 epochs the mean squared errors decrease 1000 times and a good approximation is achieved. In Figure 46B we can observe that the generalization was successful. It must be remembered that this network was able to interpolate and extrapolate well. This network was trained with a matrix with 2x54 elements and the five hidden units were enough to approximate and generalize well.

In the next example, we used twenty seven hidden neurons (see Fig. 47). This network also approximated and generalized well. The differences between this network and the one with five neurons are: the one with more neurons

converges more rapidly and generalizes better. Choosing the number of hidden

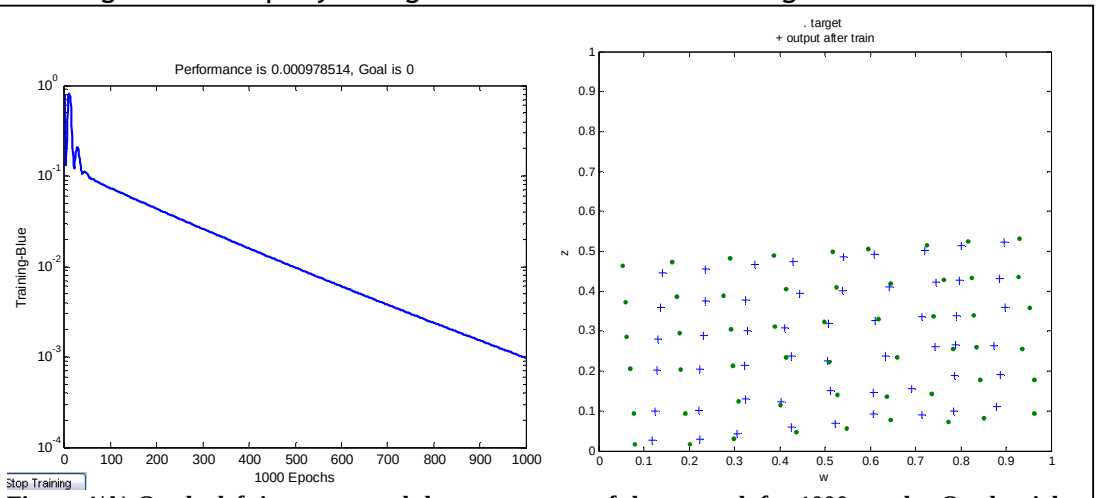


Figure 4/A) On the left is represented the convergence of the network for 1000 epochs. On the right is represented the approximating done by the network to the target.

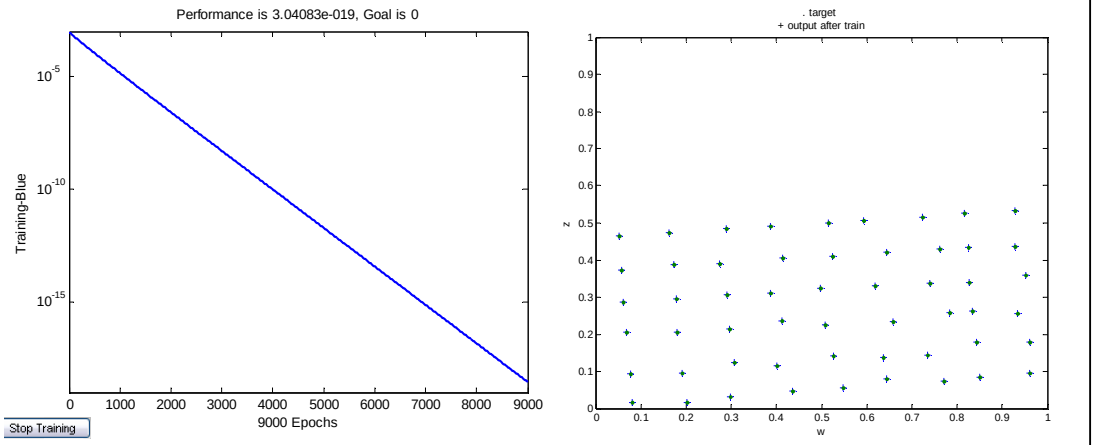


Figure 4/B) On the left is represented the convergence of the network with 27 neurons for 1000+9000 epochs. On the right is represented the approximating done by the network to the target.

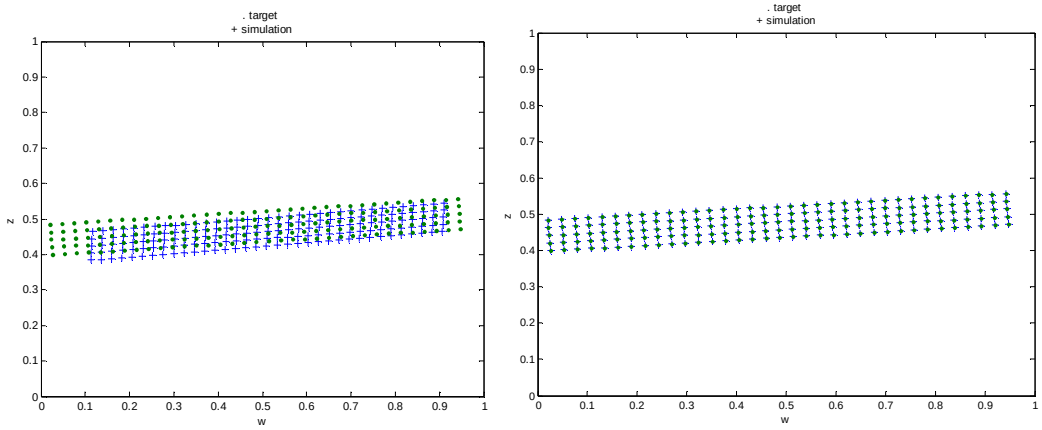


Figure 46C) On the left is represented the simulation given by the network after 1000 epochs of training. On the right is represented the simulation given by the network after10000 epochs.

neurons in a network is a very sensible problem.

It was also tested the impact of the size of the training set. Using a training set with less points (2x16) and a number of hidden neurons equal to five, it was possible to achieve a good approximation and generalization. In spite of this, the number of epochs necessary to achieve this result was 40000.

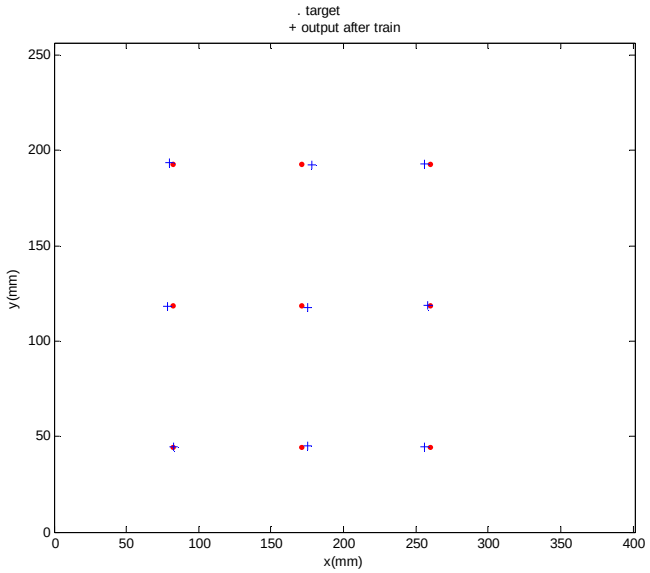


Figure 48 Comparison between the target and the output after training the network.

Returning to the data obtained with the eye tracker, due to the regularity of the patterns associated with the differences between the coordinates of the pupil and CR it was decided that the input for the network would be a matrix with two columns: the first composed by the differences between pupil and CR positions over x and the second composed by the differences between pupil and CR positions over y; the output would be a matrix with two columns where each row is the position x, y in pixels in the screen.

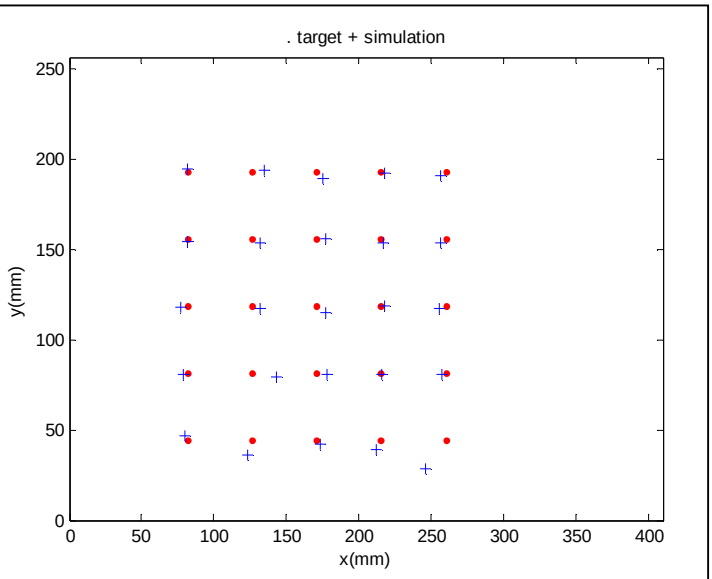


Figure 49 Comparison between the target and the output given by the network to the unseen data.

In Figure 48 we can observe the target of the network and the output after training the network. The network has three hidden neurons and two neurons in the output layer. The transfer functions are linear functions. After training the network we presented to the network, the set of points in Figure 42 in order to assess the ability for generalization. In Figure 49 is represented the output given by the network after simulation and the corresponding expected values. In Figure 50 can be observed the absolute errors associated with each coordinate, x and y respectively. These results are an estimation of the error

of the eye tracker, that is, they are an estimation of the error associated to the

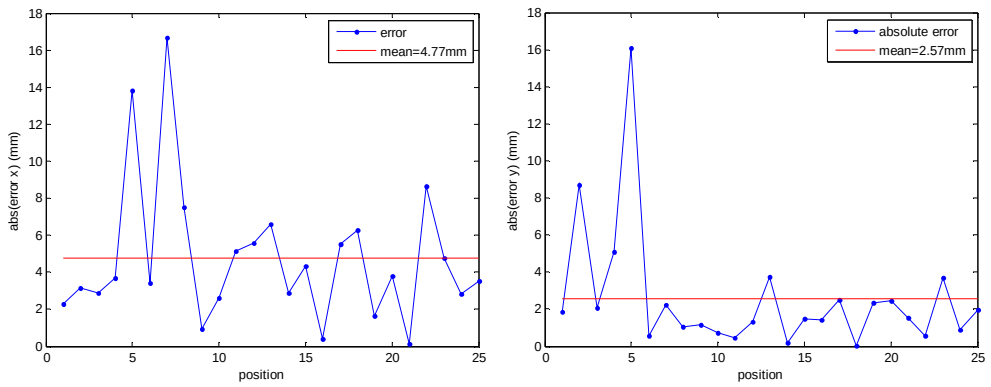


Figure 50 The errors along x(on the left) and along y(on the right) associated to the simulation of the network.

determination of the point of gaze.

In order to evaluate the performance of the eye tracker a second set of points was considered where the subject left the head fixed system and then performed the task of looking to the set with 25 points in the screen. We

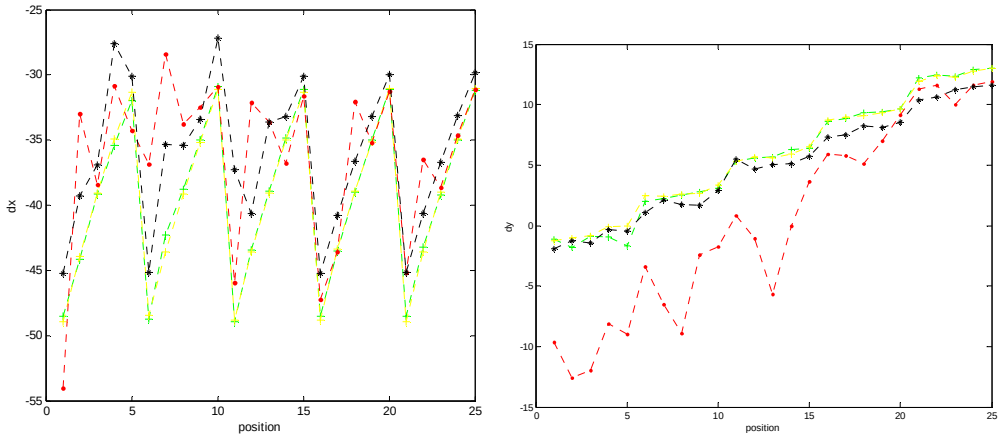


Figure 51 The differences between the pupil and CR positions, along x (left) and along y (right). The green line represents the situation previously studied (25 points). The yellow line is a repetition of the same set of points, without leaving the head fixed system and always looking to the screen. The black line is a repetition of the same set of points after leaving the head fixed system and always looking to the screen. The red line is repetition of the same set of points after leaving the head fixed system and without confining the look to the area of the screen.

observed that there was little impact on the results, in spite of this we can see less regularity for x than for y. It was also observed other aspects. While performing the task the subject may be always looking to the screen just waiting for the appearance of the next point and in this case the results show more regularity (see Figure 51). However if the subject looks to some point outside the screen, then returning to focus the points, the constancy observed for the differences along both coordinates is reduced (red line). Additionally, it was also observed that it the network was not able to generalize to data obtained with a different position of the infra red lamp.

5. CONCLUSION

The present work discussed the *active contours without edges* method in order to segment the typical images obtained with the eye tracker. It was concluded that this method was not robust and it was developed another alternative method. In the beginning, this new method showed a limitation associated to the competition between the objects in the image (the pupil and other dark areas). Since it fits to a circle, if there is another area that can contain a circle with the specified radius (sigma) it can define the circle inside that area and determine the point of minimum energy inside that area. This problem was solved using information to estimate a region of interest were the pupil is be the only dark object. This information is given by the position of the CR which has a relation with the pupil position and by the sigma which can give us an estimation of the area of the pupil. Since the CR is the lighter spot in the image, with a simple threshold it was possible to determine its position. Still, the definition of the region of interest needs to be improved since it requires the use of non automated processes, that is, it is necessary to test if the pupil's pixels are is inside the region of interest. This can be solved in a simpler but not robust way by maintaining the position of the infra red light constant for all measurements. Additionally, the use of a region of interest contributes to the decrease of the computational time that in comparison with *active contours* was reduced about 98%. The method also has the limitation of approximating the pupil's area to a circle. However, this affects accuracy but not precision. Hence, if the eye tracker is used to perform tasks which have the goal to assess the variations of the point of gaze position, this limitation is not decisive.

In relation to the calibration problem, there are some questions to be answered, such as: if the neural network that was trained with a particular set for a particular subject could generalize the calibration function for other subject; what is the number of points necessary to train the network in order to get the best generalization and flexibility; what kind of conditions (e.g. position of the infra red lamp, distance to the screen) have to be satisfied, etc. Although, these questions will only be discussed in future works, we obtained promising results that only need to be optimized. Our data shows the relation between the pupil position, the CR position and the point gaze. Additionally, we have an estimate of the eye tracker precision which is approximately 5mm for the x coordinate and 2.6mm for the y coordinate.

The work of developing the eye tracker was an opportunity to acquire a better grasp of what investigation is. This made me to perceive how challenging it can be to think creatively, autonomously and rigorously. In my opinion this work was a valuable experience within an investigation context.

6. BIBLIOGRAPHY

[1]Angelini, E., Jin, Y., Laine, A., *Segmentation of Medical Images: State of the Art of Level Set Methods in Segmentation and Registration of Medical Imaging Modalities*. pp.47-101.

[2]Böhme, M., Barth, E., *Challenges in SingleCamera Remote Eye Tracking*.

[3]Cao, F., *Geometric Curve Evolution and Image Processing*. Springer, 2002.

[4]Chan, T., Vese, L. A., *Active Contours Without Edges*. IEEE TRANSACTIONS ON IMAGE PROCESSING. Vol. 10, nº 2, pp 266-277, February 2001.

[5]Chan, T. F., Vese, L. A., *Image segmentation using level sets and the piecewise-constant Mumford-Shah model*. Kluwer Academic Publishers, 2000.

[6]DeLurgio, S. A., *Forecasting Principles and Applications*. MCGRAW-HILL INTERNATIONAL EDITIONS. pp.662-710, 1998.

[7]De Santis, A., Iacoviello, D., *Optimal segmentation of pupillometric images for estimating pupil shape parameters*. Computer methods and programs in biomedicine, nº 84, pp.174–187, 2006.

[8]Devereux, H., Eng(Hons),B., *Are Infra Red Illuminators Eye Safe?*. pp.480-481.

[9]Eastman Kodak Company, *Exploring the colour image*. KODAK Publication H-188, 1996.

[10]Haykin, S., *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.

[11]Kass, M., Witkin, A. and Terzopoulos, D., *Snakes: Active contour models*, Int. J. Comput. Vis.. Vol. 1, pp. 321–331, 1988.

[12]Lawrence, S., Giles, C. and Tsoi, A., *Lessons in Neural Network Training:Overfitting May be Harder than Expected*. Proceedings of the Fourteenth National Conference on Artificial Intelligence, pp. 540-545, 1997.

[13]Lee, C. P., *A dissertation submitted to the Graduate Faculty of North Carolina State University in partial satisfaction of the requirements for the Degree of Doctor of Philosophy*. Raleigh, 2005.

[14]Moireau, P., Fliss, S., *Segmentation d'images par contours actifs sur le modèle de Mumford-Shah*. Janvier - Avril 2004

[15]Osher, S., Fedkiw, R. P., *Level Set Methods: An Overview and Some Recent Results*. September 5, 2000.

[16]Osher, S., Sethian, J. A., *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*. Journal of Computational Physics, vol. 79, pp 12–49, 1988.

[17]Pratt, W. K., *Digital Image Processing: PIKS Inside*. John Wiley & Sons, Inc., 2001

[18]Richardson, D., Spivey, M., *Eye-Tracking: Characteristics and Methods*. Encyclopedia of Biomaterials and Biomedical Engineering, Wnek. G. & Bowlin, G., 2004.

[19]Rumelhart, D, *Parallel Distributed Processing*. Mit Pr,1986

[20]Sethian, J. A., *Evolution, Implementation, and Application of Level Set and Fast Marching Methods for Advancing Fronts*. 2001, Review article

[21]Stone, M., *The Generalized Weierstrass Approximation Theorem*. Mathematics Magazine, Vol. **21**, No. 4, pp-167-184, 1948

[22]Tetko, I., Livingstone, D. and Luik, *A. Neural Network Studies. 1 Comparison of Overfitting and Overtraining*. J. Chem. Inf. Comput. Sci., 35, pp. 836-833, 1995

[23]Young, I. T., Gerbrands, J., van Vliet, L., J., *Fundamentals of Image Processing*, 1995.